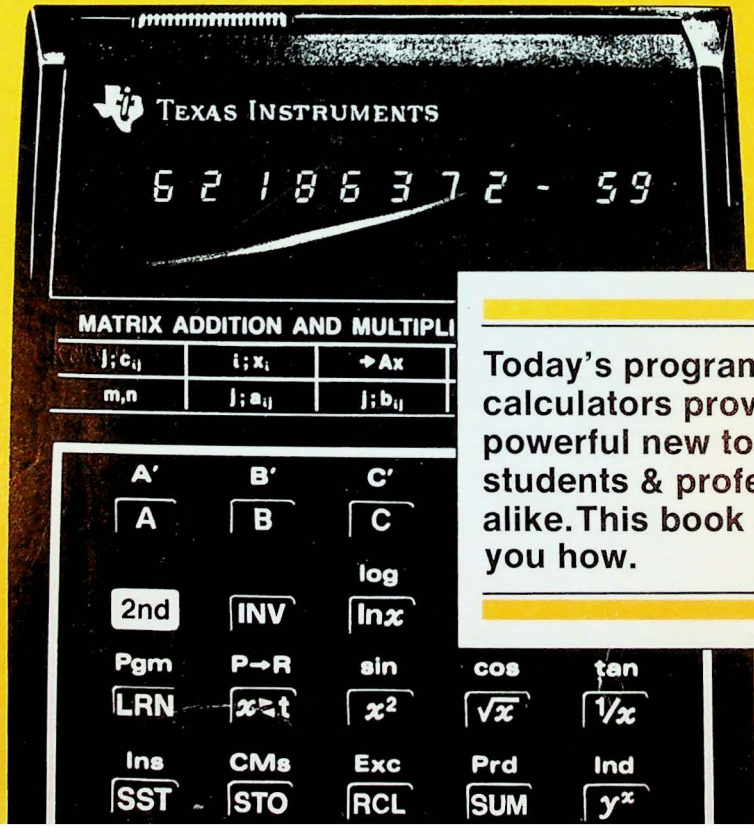
$12.95

Sourcebook for Programmable Calculators

Step-by-step programmed solutions, whatever your field: ■ mathematics ■ calculus ■ statistics ■ business and operations research ■ economics ■ biology ■ engineering ■ physics and astronomy ■ music ■ and much more . . . designed for use with the TI Programmable 58 and 59 calculators



Today's programmable calculators provide a powerful new tool for students & professionals alike. This book shows you how.

This book was developed by:

The Staff of the Texas Instruments Learning Center
Jacquelyn F. Quiram
Diana M. Koperski
Dr. Ralph A. Oliva, Educational Software Director

And

The Staff of Rensselaer Polytechnic Institute
Dr. Robert Resnick, Program Director, Professor of Physics and
E. P. Hamilton Distinguished Professor of Science Education

With contributions by

Dr. A. Bruce Carlson, Associate Professor of Electrical Engineering
Dr. Richard A. Harper, Associate Professor of Physics
Dr. Karl W. Heiner, Assistant Professor of Operations Research and Statistics
Dr. Lee E. Ostrander, Associate Professor of Biomedical Engineering
Dr. Richard L. Smith, Assistant Professor of Physics
Carol B. Thompson, Research Associate in Management
Dr. James A. Voytuk, Associate Professor of Mathematics
Dr. John W. Wilkinson, Professor of Management
Dr. Alexej Wynnyczuk, Professor of Economics

Additional Contributions by

Roger F. Farish
M. Dean LaMont
Robert E. Lee, Instructor
School of Education and Allied Professions, Miami University
Michael R. Zastrocky

Other Contributions by

Milton F. Baldwin
Dr. Elbert B. Greynolds, Jr.
Delores A. Hackney
Judith A. Lipsett
Gerald Luecke
Dr. Richard G. Moore
Arthur L. Norrington, Jr.
Joe E. Poyner
James F. Reed
Cathy Tiberia

Artwork and Layout were coordinated and executed by

Schenck, Plunk & Deason

ISBN 0-89512-025-9
Library of Congress Catalog Number 78-57030

Copyright © 1978 by Texas Instruments Incorporated

Table of Contents

INTRODUCTION

Calculator Evolution i-1
Calculator Roles in Today's College Studies i-3
Programmability i-4
Some Specifics i-4
How to Use This Book i-6
Notes About Your Calculator i-7

1. BASIC NUMBER THEORY

Introduction 1-1
Sequences and Series 1-2
Greatest Common Divisor 1-6
Prime Number Testing 1-9
Prime Factorization 1-12
Base Conversion 1-16

2. COLLEGE ALGEBRA AND TRIGONOMETRY

Introduction 2-1
Graphing Functions 2-2
Polynomial Evaluations 2-7
Systems of Linear Equations (Master Library Program) 2-10
Quadratic Equations 2-15
Trig and Triangles 2-19
Trig Functions 2-22

3. CALCULUS AND LINEAR ALGEBRA

Introduction 3-1
Approximating Roots of a Function 3-2
Numerical Approximation of Derivatives 3-6
Definite Integrals—Trapezoidal Rule 3-10
Runge-Kutta Method (Systems of Differential Equations) 3-14
Curve-Fitting with Least-Squares Approximation 3-26
Systems of Linear Equations (Row Reduction) 3-33

4. STATISTICS

Introduction 4-1
Descriptive Measures 4-2
The Normal Curve 4-10
The Standard Normal Curve 4-15
t-Distribution 4-19
Linear Regression and the Correlation Coefficient 4-24
Analysis of Variance 4-31
Tables 4-36

5. MUSIC THEORY

Introduction 5-1
Intervals 5-3
Chord Identification 5-8
Sight-Singing and Melodic Contours 5-13
Composition 5-20
Pitch Frequency 5-26
Harmonics 5-30
Selected Listening Guide for Computer and Electronic Music 5-33

Table of Contents

6. Business and Operations Research

Introduction . . . 6-1
Inventory Control . . . 6-2
Variable Periodic Cash Flows and Internal Rate of Return . . . 6-5
Capital Budgeting . . . 6-12
Queuing Model . . . 6-20
Markov Analysis . . . 6-25

7. Economics

Introduction . . . 7-1
Seasonal Index (Ratio to Centered Average) . . . 7-3
Reference Cycle . . . 7-9
Fluctuation Around the Trend Line . . . 7-18
Interaction of Multiplier and Accelerator . . . 7-23
Short-Run Production Function and Cost Curves . . . 7-28

8. Biology

Introduction . . . 8-1
Genetics . . . 8-2
Blood-Type Matching . . . 8-8
Population Growth . . . 8-15
Sedimentation . . . 8-21

9. Biomedical Engineering

Introduction . . . 9-1
Active Filter to Improve Signal-to-Noise Ratio . . . 9-3
Estimating Number of Samples for Specified Standard Error . . . 9-7
Transducer Scaling and Interactions . . . 9-10
Cardiac Output Using the Fick Equation . . . 9-18
Student's t-Test . . . 9-24
Regression Analysis . . . 9-30

10. Electrical and Systems Engineering

Introduction . . . 10-1
Series-Parallel Impedance . . . 10-2
Smith Chart Calculations . . . 10-6
Transistor Parameter Conversion . . . 10-16
Bode/Nyquist Calculations . . . 10-21
Root-Locus Calculations . . . 10-28
Step and Pulse Response . . . 10-35
Convolution . . . 10-39
Discrete-Time System Response . . . 10-44

11. Physics and Astronomy

Introduction . . . 11-1
Fraunhofer Diffraction at a Circular Aperture . . . 11-2
Euler's Method and Studies of Motion . . . 11-8
Central Force Motion . . . 11-16
Two-Dimensional Speed Distribution . . . 11-25
The Equation of Time . . . 11-37
Sundial Design . . . 11-43

Index

Introduction

Modern mathematics is a many-faceted subject required in a wide range of studies and careers. Analyzing and testing data, investigating relationships, and applying mathematical models to real-life situations — these are skills that are essential to many disciplines, and they can offer you exciting and stimulating insights into the way the world operates. But certain parts of mathematics — compiling data, checking and rechecking calculations and performances, the "arithmetic" part of mathematics — can be quite tedious. You've probably looked for ways to increase the speed and accuracy of calculations like these.

Calculator Evolution

You are not alone. Throughout the centuries mankind has searched for tools to help perform the arithmetic side of mathematics more easily and quickly. Your programmable calculator is another link in what has been and will continue to be an explosive evolution in the development of better, faster, and more accurate computational tools.

Historians usually consider the abacus, which originated in the Orient as early as the 6th century B.C., to be the first real calculating device. The abacus has existed in various forms in many cultures and is still in use in some areas today. For many centuries there was little improvement in these basic tools, but by the 17th century the evolution had begun.

The invention of the first actual calculating machine is credited to the French scientist-philosopher Blaise Pascal in 1642, although work along these lines had been done by earlier inventors and innovators. By 1671 Gottfried Leibniz had developed his more advanced machine, the "Stepped Reckoner," which could add, multiply, divide, and extract square roots. Early attempts to manufacture calculators, however, produced highly unreliable results. It was not until 1820 that the first commercial machines handling addition, subtraction, multiplication, and division became available.

Another advanced idea occurred during the 1800's. Punched-card automation systems were invented to drive mills and looms, like the one invented by Joseph Marie Jacquard. An Englishman, Charles Babbage, saw the possibilities of data storage on punched cards for a calculating device. He proposed building an "Analytical Engine" that would store both numbers *and* the operations to be performed with them, use the results of one calculation as input data for another, and perform repetitive computations (loops) and conditional transfers. Although Babbage was never able to complete a working prototype, his designs stand as a milestone in the evolution of today's programmable calculators and computers. As he predicted, it was a half-century after his death in 1871 before his visionary goals were accomplished.

Calculator Evolution

Until the late 19th century all computing devices were operated mechanically by gears, ratchets, springs, cranks, and levers. Herman Hollerith was the first to add an electric card reader to data processing. His system was used to tabulate the 1890 U.S. census, and in 1896 he entered the commercial market. His company was called the Tabulating Machine Company; in 1924, after several mergers, it became the International Business Machine Corporation.

Electromechanical calculators held the field from 1925 until the middle 1940's, culminating in the Mark I computer completed by Harvard University in 1944. Even before it was finished, however, work had started on the first electronic computer, the ENIAC.

The rapidly growing electronic computer technology quickly outstripped the performance of electromechanical and electrical calculating devices. The ENIAC was more than 1,000 times faster than its electromechanical counterparts. Its vast array of circuitry and vacuum tubes was contained in thirty units, requiring a space of 30 by 50 feet and weighing more than 30 tons. Clearly, ways had to be found to reduce the size, complexity, and cost of computers.

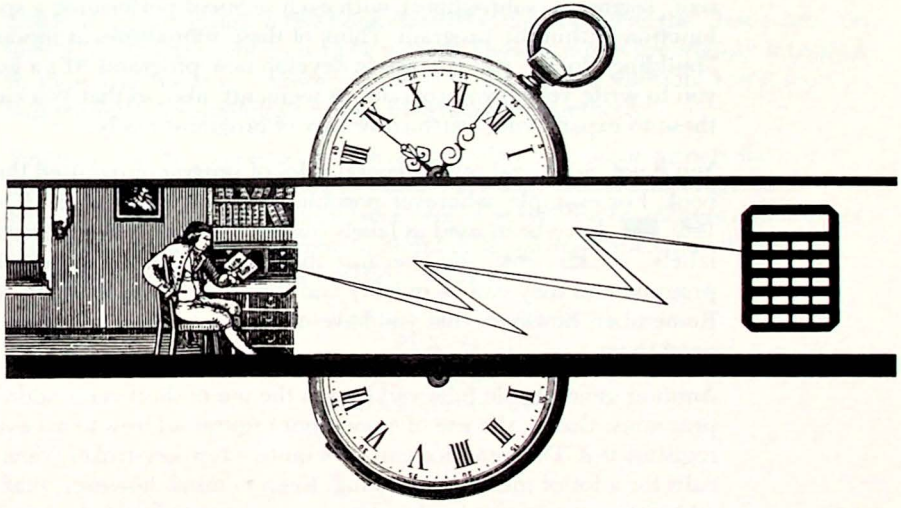
During the 1950's, computer technology searched for new refinements and advances toward this goal. Transistors, which replaced the vacuum tube, greatly reduced the size and cost of computers and increased their reliability. But an even more exciting breakthrough came in 1958 through the work of Jack Kilby at Texas Instruments. That breakthrough was the integrated circuit.

The integrated circuit or "IC" is a tiny "chip" of silicon with complete electronic circuits, including thousands of transistors, diodes, capacitors, resistors, and their required interconnections, built into it by a series of techniques resembling photographic development. Today's sophisticated IC's can perform calculations that required a room-sized computer 25 years ago. Along with companion developments in various display technologies, these tiny devices have been a key factor in making handheld calculators a practical and affordable reality.

Calculator Roles in Today's College Studies

So welcome to a new era — the era of "calculator analysis." Modern technology has placed a new dimension in instant problem-solving capability in your hands. The power and versatility of calculators such as the TI Programmable 58 and TI Programmable 59 offer you not only easy and accurate computation, but also another important commodity: more time. Time to explore relationships in data. Time to try out mathematical models that were previously too complex and time-consuming to approach in the classroom. Time to look at more of the "what-if's" in any problem-solving situation.

Modern education is beginning to feel the impact of calculators in the classroom. From teaching and reinforcement of basic math concepts in the primary grades to advanced graduate-level research, these compact, powerful machines can help to make the job easier for students and teachers alike. Chances are that you have already owned at least one handheld calculator that performed basic arithmetic and allowed you to concentrate more on the analytical side of mathematics. Now, your programmable calculator can open whole new avenues of enjoyment and creativity for you.



PROGRAMMABILITY

PROGRAMMABILITY

Programmability greatly expands the power and usefulness of your calculator. Routines that previously could be attempted only on a computer can now be performed right at your fingertips. Iterative calculations that used to require hours will essentially take little more time than is needed to enter your program into the program memory.

You won't need to go through a lengthy training period or learn super-sophisticated skills in order to use the programmable capabilities of your calculator. Basically, you simply tell the calculator what you want it to do and the order in which the operations are to be done. Chapter Four of *Personal Programming*, the owner's manual for the TI Programmable 58/59, takes you step by step through the basics of programming your calculator, and you'll soon become comfortable with these techniques, based largely on common sense and logical thinking. The small amount of time you'll spend mastering the use of your calculator can yield enormous and continuous returns.

SOME SPECIFICS

Included here are some of the guidelines that were followed in preparing the programs in this book. For the most part, the programs are written in "bite-size" segments (subroutines), with each segment performing a specific function within the program. Think of these subroutines as modules or "building blocks" you can use to develop new programs. It's a good idea for you to write your own programs in segments, also, so that you can draw on these to expand your personal library of programs easily.

You'll notice several other general rules of programming used throughout this book. For example, wherever possible the user-defined keys [A] through [2nd] [E'] have been used as labels (rather than what are called the "common labels" – [x^2], [ln x], etc.) because they are more easily identifiable within a program and they can be quickly and simply accessed from the keyboard. Remember, however, that you have many more labels to call on when you need them.

Another general rule followed here is the use of short-term addressing within programs; that is, the use of a one-digit register address to access data registers 0-9. This practice can save quite a few keystrokes when a program calls for a lot of memory accessing. Keep in mind, however, that short-form addressing *can be used only when a non-numeric keystroke immediately follows the register address.*

Generally speaking, absolute addressing and indirect addressing have been avoided, to keep the programs as straightforward as possible. You will find a few examples of absolute addressing in programs that would require a very long performance time without it. Indirect storage and recall have been used occasionally when it was necessary to access memory registers sequentially within a program.

Print commands are written into the programs so that results will automatically be printed if the PC-100A thermal printer is used. If you are using the calculator alone, these commands are ignored by the calculator as a program is run. (You may choose to omit them entirely when keying in the program.)

For more information on these techniques, refer to these pages in *Personal Programming:*

Subroutines — pp. IV-46ff, V-58ff.
Labels — pp. IV-11, IV-43, V-55, and V-56.
Short-form addressing — IV-15, IV-44, V-22, and V-58.
Absolute addressing — IV-44, IV-86, V-57.
Indirect addressing — IV-84, IV-86, V-68.
Printer Control — Chapter VI.

None of the programs in this book should be considered as the "last word" on how a particular problem is to be solved. These samples are merely intended to show you one way — among many — to approach a solution. You'll soon find yourself developing your own programming methods.

You may also be interested in seeing how other calculator owners have approached situations similar to yours. PPX-59 (Professional Program Exchange) offers a wide selection of programs written by owners and users of the TI Programmable 59. Annual membership dues at the time of this printing are $15.00. For information, write to:

Texas Instruments Incorporated
PPX-59
P.O. Box 53
Lubbock, Texas 79408

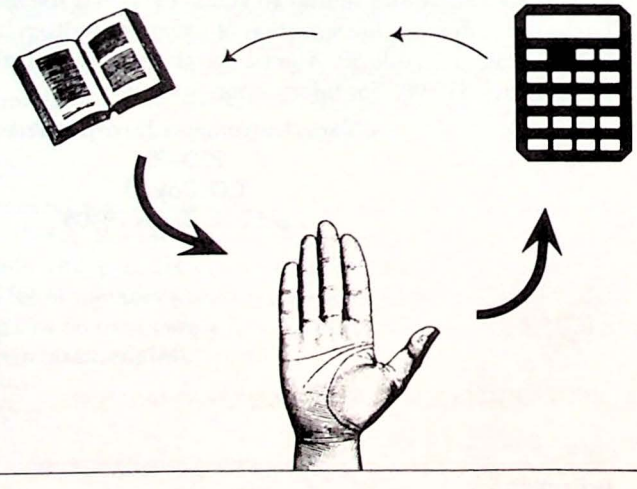
How to Use This Book

This book is designed to guide you into the application of calculator analysis either as part of your studies or in your work. Of course, it does not attempt an exhaustive or totally comprehensive review of all the uses of a programmable calculator in every discipline. Instead, we hope that the examples included here will stimulate and challenge you to create your own programs and to find new ways to use your calculator in your classes and career.

Several of the topics covered here are common to many disciplines. Since basic number theory, college algebra and trigonometry, calculus, and statistics are required in many degree programs, these chapters are grouped at the beginning of the book. The remaining chapters offer sample applications in music, business, economics, biology, biomedical engineering, electrical engineering, and physics and astronomy.

Each chapter contains a special selection of example situations, along with a brief look at the theory and background necessary to lead you to a program solution. A sample solution is included in each case, and related examples are outlined for you to explore on your own.

For convenience, the material in this book is cross-referenced to *Personal Programming* and the *Master Library* manual wherever necessary. Remember, too, to consult the index in the back of the book for programs that can be of use to you.



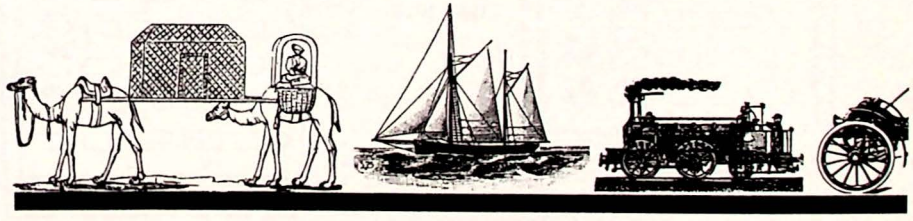
NOTES ABOUT YOUR CALCULATOR

With minor exceptions the programs in this book have been designed for both the TI Programmable 58 and TI Programmable 59. A few programs have too many steps to be entered in entirety into the TI Programmable 58. In most of these, however, the major subroutines can still be used, with intermediate calculations performed directly from the keyboard.

The major differences in the two calculators lie in the maximum number of data and program storage registers available and the magnetic card reading and writing capability of the TI Programmable 59. The differences are summarized in the following chart.

	TI Programmable 58	*TI Programmable 59*
Partitioning when first turned on	30←memory registers→60 240←program steps→480	
Maximum data storage registers (memories)	60 (0 program steps)	100 (160 program steps)
Maximum program steps	480 (0 memories)	960 (20 memories)
Magnetic card reading and writing	no	yes

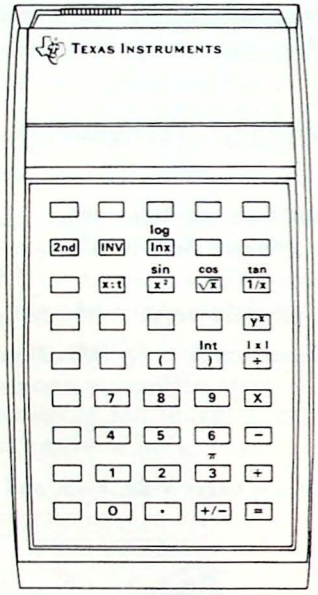
For a full discussion of partitioning, see pages V-22, V-29, and V-42 in *Personal Programming*.



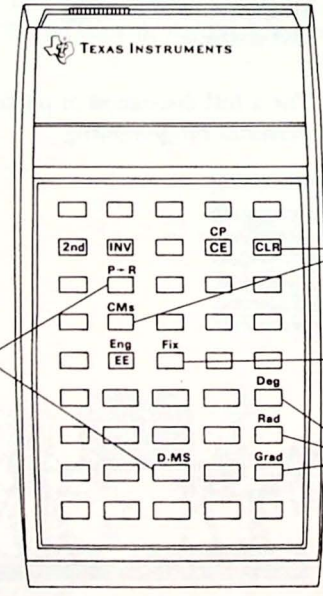
Key Arrangement Diagrams

The keys of the TI Programmable 58 and 59 are conveniently grouped according to their function. The following diagrams review the key arrangement and are cross-referenced by chapter and/or page number to the sections in *Personal Programming* that discuss the functions of the keys.

Data Entry, Basic Operation, and Algebraic Function Keys: Chapters II, V



Special Operation Keys



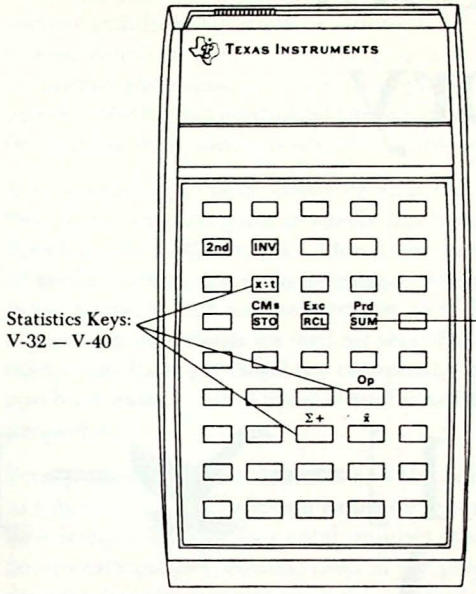
Clearing Keys: II-2, II-6

Display Format Keys: II-8, II-9

Angular Mode Keys: II-12

Conversion Keys: II-13, II-14

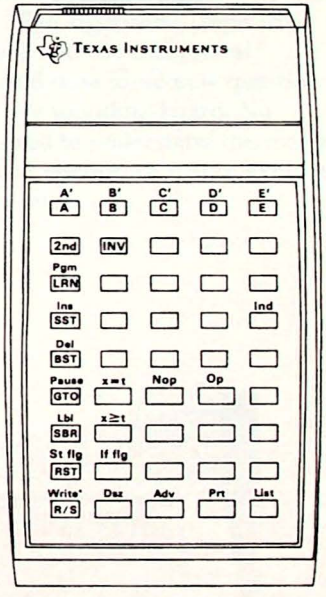
Memory and Statistics Keys



Statistics Keys:
V-32 – V-40

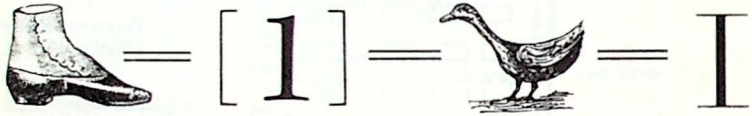
Memory Keys:
II-6, II-7

Programming Keys:
Chapters IV, V, VI, VII



*The TI Programmable 58™ does not have this function.

1 Basic Number Theory

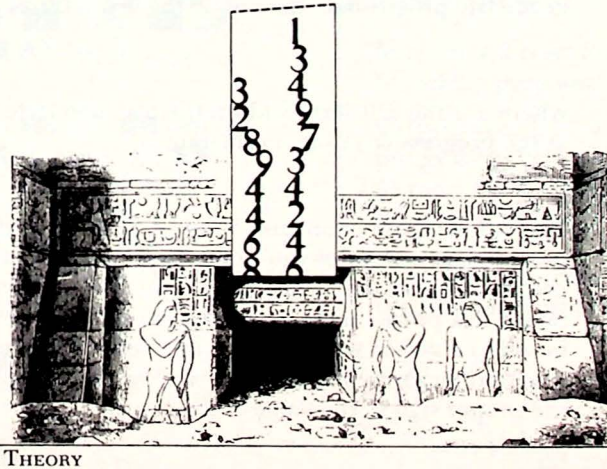


Introduction

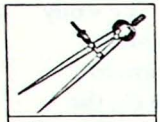
The properties of the integers have fascinated people for centuries. The study of numbers can be traced back as early as 1600 B.C., when the Babylonian mathematicians computed various numerical properties out of a desire to understand the magnitude of numbers. Then, beginning about 300 B.C., the Greeks developed the foundations for a systematic study of numbers based on axioms and supported by logical proofs. The axiomatic approach to number theory has continued through the centuries, and even today this fascination with numbers occupies a major place in mathematical research.

It is interesting to note, however, that the computational aspect of number theory, rather than pure research, has been the driving force in the development of the subject. One does not need to understand a large amount of mathematical theory to investigate the properties of numbers. In fact, many properties of number systems were first formulated by actually computing the results for various sets of numbers. Modern high-speed computers have provided the computing power to justify many conjectures and have even in some cases actually assisted in proving number theory properties.

Your programmable calculator and the computational techniques presented in this chapter can help you examine the properties of numbers and develop new insights into fundamental number theory. The algorithms (repetitive computational procedures) used in the programs and the theoretical discussions which accompany the programs could pose some new questions, for which the calculator may serve as an effective sounding-board. No sophisticated mathematical knowledge is required to understand the material included in this chapter, and we hope that it will encourage you to explore the properties of numbers even further on your own.

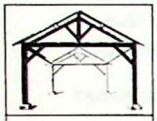


1 Sequences and Series



OBJECTIVE

Sequences and series are important mathematical concepts which weave themselves through calculus, probability, statistics, finance, and many other mathematical endeavors. The following program can help you to analyze and evaluate various sequences by finding the nth term and the sum of the first n terms of an arithmetic or geometric progression.



FACTS/ THEORY

Generally a *sequence* is a set of numbers that are in a one-to-one correspondence with the set of positive integers n. The sequence may be finite or infinite. The numbers or members of the set are called terms.

A *series* is closely related. A simple way to think of a *series* is to describe it as a sequence with the symbol + between each term. For example, 2, 4, 6, 8, 10, . . . is an infinite sequence, while $2+4+6+8+10+\ldots$ is an infinite series.

An *arithmetic progression* is a sequence in which each term of the sequence is the sum of the preceding term and a constant. For example, 2, 4, 6, 8 . . . is an arithmetic progression where the fixed constant is 2.

You can easily find the sum of the first n terms of an arithmetic progression (S_n) by evaluating the formula

$$S_n = n/2[2a + (n-1)k]$$

where a is the first term and k is the constant. To find the nth term of the progression (t_n), you evaluate the formula

$$t_n = a + (n-1)k$$

A *geometric progression* is a sequence in which each term is the product of the preceding term and a constant multiplier (common ratio). For example: 4, 2, 1, ½, . . . is a geometric progression where the common ratio is ½. For a geometric progression, the sum of the first n terms (S_n) is

$$S_n = \frac{a - ar^n}{1 - r}$$

where a is the first term and r is the common ratio. You can find the nth term of the progression (t_n) by evaluating

$$t_n = ar^{n-1}$$

The following program finds S_n and t_n for arithmetic and geometric progressions and helps you explore aspects such as convergence. Note that the program checks the value of r to see if it is negative or positive, since it is raised to a power in the geometric progression formulas. A negative value raised to a power with the y^x key causes a flashing display; therefore the program performs an odd/even test on the exponent (if $r<0$) and assigns the proper sign to the value of r^n or r^{n-1}.

THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [A] [STO] 1 [R/S]	Stores a in memory 1.
005	[2nd] [Lbl] [B] [STO] 2 [2nd] [x≥t] [2nd] [E'] [2nd] [St flg] 1 [R/S]	Stores r (or k) in memory 2 and sets flag 1 if r is negative. If r is 0 or positive, program transfers to [E'] and stops.
014	[2nd] [Lbl] [C] [STO] 3 [R/S]	Stores n in memory 3.
019	[2nd] [Lbl] [2nd] [E'] [INV] [2nd] [St flg] 1 [R/S]	If n is positive, the [INV] [2nd] [St flg] 1 insures that flag 1 is not set.
025	[2nd] [Lbl] [D] [RCL] 1 [+] [(] [RCL] 3 [−] 1 [)] [×] [RCL] 2 [=] [2nd] [Prt] [R/S]	Calculates t_n for an arithmetic progression.
042	[2nd] [Lbl] [E] [RCL] 3 [÷] 2 [×] [(] 2 [×] [RCL] 1 [+] [(] [RCL] 3 [−] 1 [)] [×] [RCL] 2 [=] [2nd] [Prt] [R/S]	Calculates S_n for an arithmetic progression.
067	[2nd] [Lbl] [2nd] [A'] [2nd] [St flg] 2 [RCL] 1 [×] [RCL] 2 [2nd] [|x|] [y^x] [(] [RCL] 3 [−] 1 [)] [STO] 4 [=] [STO] 6 [2nd] [If flg] 1 [2nd] [C'] [2nd] [Prt] [R/S]	Calculates t_n for a geometric progression ($a \times |r|^{n-1}$ stored in memory 6). If r is negative, program transfers to [C']. *Note:* Flag 2 is set for later branching purposes.
094	[2nd] [Lbl] [2nd] [B'] [INV] [2nd] [St flg] 2 [RCL] 2 [2nd] [|x|] [y^x] [RCL] 3 [STO] 4 [=] [STO] 6 [2nd] [If flg] 1 [2nd] [C'] [2nd] [Lbl] [x^2] [RCL] 6 [×] [RCL] 1 [=] [+/−] [+] [RCL] 1 [=] [÷] [(] 1 [−] [RCL] 2 [)] [=] [2nd] [Prt] [R/S]	Calculates S_n for a geometric progression ($|r|^n$ stored in memory 6). If r is negative, program transfers to [C']. *Note:* Flag 2 is unset for later branching purposes.

Step #	Program Keystrokes	Purpose/Comments
136	2nd Lbl 2nd C' RCL 4 ÷ 2 = 2nd Int × 2 = − RCL 4 = INV 2nd x≥t 2nd D' INV 2nd If flg 2 x² RCL 6 2nd Prt R/S	Recalls value of n or n-1 and checks to see if it is odd or even. If odd, program transfers to D'. If even, program checks to see if flag 2 is set or unset. If flag 2 is not set, program transfers to subroutine x². If flag 2 is set, program recalls ar^{n-1} from memory 6 and stops.
162	2nd Lbl 2nd D' RCL 6 +/− STO 6 INV 2nd If flg 2 x² 2nd Prt R/S	Changes sign of quantity in memory 6 and checks again to see if flag 2 is set or unset. If not set, program transfers to x²; if set, program stops.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

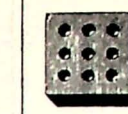
REGISTER CONTENTS

R_{00}	R_{03} n	R_{06} Used	R_{09}
R_{01} a	R_{04} Used	R_{07}	
R_{02} k or r	R_{05}	R_{08}	

USING THE PROGRAM

Procedure	Press	Display/Comments
For arithmetic progression:		
Enter a.	A	a
Enter k.	B	k
Enter n.	C	n
Calculate t_n.	D	t_n °
Calculate S_n.	E	S_n °
For geometric progression:		
Enter 0 in t register.	0 x:t	0.
Enter a.	A	a
Enter r.	B	r
Enter n.	C	n
Calculate t_n.	2nd A'	t_n °
Calculate S_n.	2nd B'	S_n °

°This value is printed if the PC-100A is used.



SAMPLE PROBLEM

(1) Find t_n and S_n for the arithmetic progression with $a = 4$, $k = -2$, and $n = 12$.
(2) Find t_n and S_n for the geometric progression with $a = 2$, $r = 3$, and $n = 6$.

Procedure	Press	Display/Comments
First, enter program carefully.		0
(1)		
Enter the value a.	4 [A]	4.
Enter the value k.	2 [+/−] [B]	−2.
Enter the value n.	12 [C]	12.
Compute t_n.	[D]	−18.
Compute S_n.	[E]	−84.
(2)		
Enter 0 in t register.	0 [x:t]	0.
Enter the value a.	2 [A]	2.
Enter the value r.	3 [B]	3.
Enter the value n.	6 [C]	6.
Compute t_n.	[2nd] [A']	486.
Compute S_n.	[2nd] [B']	728.

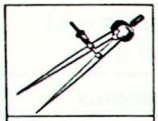
FURTHER EXPLORATIONS

1. Find t_n and S_n in the arithmetic progression where $a = 5$, $k = 3/8$, and $n = 45$. (*Answer:* $t_n = 21.5$, $S_n = 596.25$)
2. The first term of an arithmetic progression is 4 and the last term is 424. Find the sum of the sequence if $k = 2$. (*Answer:* $n = 211$, $S_n = 45154$)
3. As n gets larger, what value does S_n seem to approach for the geometric progression with $a = 4$ and $r = 1/2$? (*Answer:* 8)
4. As n gets larger what value does t_n seem to approach for the geometric progression with $a = 41$ and $r = .01$? (*Answer:* 0)

REFERENCES/ INFORMATION

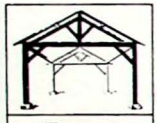
Most basic mathematics texts include a discussion of sequences and series. See, for example, Charles D. Miller and Vern E. Heeren, *Mathematics: An Everyday Experience* (Glenview, Illinois: Scott, Foresman and Company, 1976), chapter three.

1 Greatest Common Divisor



OBJECTIVE

Given two integers a and b, a common divisor of a and b is an integer which divides both numbers, and the greatest common divisor (g.c.d.) is the largest such integer. The g.c.d., although not a particularly complicated concept in algebra and number theory, is used repeatedly to define properties and impose restrictions on results. In addition to the actual computation of the g.c.d., the procedure outlined in this section serves as a good demonstration of a computational algorithm, which, when systematically performed, yields the result.



FACTS/ THEORY

The most well-known algorithm for computing the greatest common divisor, and perhaps one of the oldest, is the Euclidean Algorithm, found in Book VII of Euclid's "Elements." Given positive numbers a and b ($a>b$), the algorithm proceeds as follows:

1. Divide a by b and obtain remainder r_1.

$$a = bq_1 + r_1 \qquad 0 \le r_1 < b$$

Note that every integer that divides a and b will also divide r_1, and, conversely, every common divisor of b and r_1 is a divisor of a. Therefore, the common divisors of a and b are the same as those of b and r_1, and it follows that the g.c.d. of a and b equals the g.c.d. of b and r_1.

2. Now divide b by r_1 and find r_2.

$$b = r_1q_2 + r_2 \qquad 0 \le r_2 < r_1$$

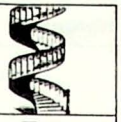
The above remarks hold for the relationship between the pairs b, r_1 and r_1, r_2, so you continue the procedure until the remainder, r_n, is zero.

$$\begin{aligned}
a &= bq_1 + r_1 && 0 \le r_1 < b \\
b &= r_1q_2 + r_2 && 0 \le r_2 < r_1 \\
r_1 &= r_2q_3 + r_3 && 0 \le r_3 < r_2 \\
&\bullet\ \bullet\ \bullet \\
r_{n-3} &= r_{n-2}q_{n-1} + r_{n-1} && 0 \le r_{n-1} < r_{n-2} \\
r_{n-2} &= r_{n-1}q_n && (r_n = 0)
\end{aligned}$$

Because the remainder is always decreasing, it must eventually vanish. Since r_{n-1} is a divisor of r_{n-2} and itself, it is the g.c.d. of all pairs r_{n-2}, r_{n-1}; r_{n-3}, r_{n-2}; . . . back to a, b.

Once the g.c.d. of two numbers has been found, the least common multiple (the smallest number divisible by both a and b) can be easily computed. If d is the g.c.d. of a and b, then the least common multiple of a and b is

$$m = ab/d$$



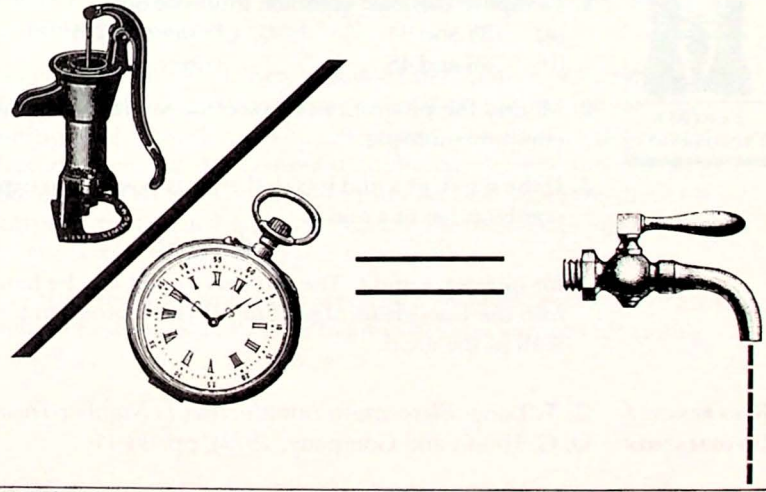
THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [A] [STO] 1 [R/S]	Stores a in memory 1.
005	[2nd] [Lbl] [B] [STO] 2 [R/S]	Stores b in memory 2.
010	[2nd] [Lbl] [C] [RCL] 1 [÷] [RCL] 2 [=] [2nd] [Int] [×] [RCL] 2 [+/−] [+] [RCL] 1 [=] [2nd] [Exc] 2 [STO] 1 [RCL] 2 [x≥t] 0 [INV] [2nd] [x=t] [C] [RCL] 1 [2nd] [Prt] [R/S]	Divides a by b and finds r_1. Advances b to memory 1 and stores r in memory 2. Then tests r to see if r = 0; if not, program loops back to [C] and repeats until r = 0. Program then calls r_{n-1} (g.c.d.) from memory 1 and stops.
	[LRN] [RST]	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}
R_{01} a
R_{02} b



1

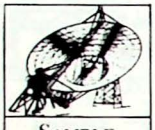
Greatest Common Divisor

Using the Program

Procedure	Press	Display/Comments
Enter a.	A	a
Enter b.	B	b
Calculate g.c.d.	C	g.c.d.*

*This value is printed if the PC-100A is used.

Note that the time it takes to find the g.c.d. depends on how many times the program must loop through the iterative routine.



Sample Problem

1. Compute the greatest common divisor of 252 and 180.

Procedure	Press	Display/Comments
Enter program.		
Enter a.	252 A	252.
Enter b.	180 B	180.
Calculate g.c.d.	C	36. g.c.d.

2. Compute the greatest common divisor of 3331 and 729.

Procedure	Press	Display/Comments
Enter a.	3331 A	3331.
Enter b.	729 B	729.
Calculate g.c.d.	C	1. (3331 is prime.)

Further Explorations

1. Compute the least common multiple of:
   (a) 128 and 94. (*Answer:* 6016)
   (b) 756 and 48. (*Answer:* 3024)
2. Modify the program in this section to yield both the g.c.d. and the least common multiple.
3. If the g.c.d. of a and b is d, then it is possible to express d as a linear combination of a and b:

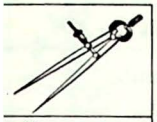
$$d = sa + tb$$

for integers s and t. The integers s and t can be found by back substitution into the Euclidean Algorithm. Write a program for computing s and t as well as the g.c.d.

References/ Information

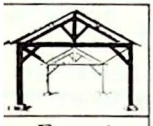
C. T. Long, *Elementary Introduction to Number Theory* (Lexington, Mass.: D. C. Heath and Company, 1972), pp. 33-41.

Prime Number Testing



OBJECTIVE

The question of whether or not a given integer N is prime or composite has not been solved, and only partial answers have been obtained. It was proved by Euclid (about 300 B.C.) that there are infinitely many primes. It is also possible to show that the primes occur less and less frequently among the larger integers, with estimates on the density of occurrence. However, the computation or testing of primes is still of interest, and new large computers have been employed to generate larger and larger primes. Aside from answering questions about the factorization and divisors of integers, the investigations are purely intellectual, but problems of this kind are still of interest to many people.



FACTS/ THEORY

The computation and testing problem to determine prime numbers has led to the discovery of many interesting properties for the natural numbers. Investigations include estimates for the number of primes less than a large number N, formulas which yield primes for certain sequences of integers, and the spacing between consecutive primes.

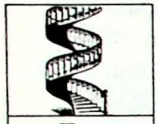
A simple and straightforward scheme for determining if a number N is prime is one called the Sieve of Eratosthenes, after the Greek mathematician, 276-194 B.C. The method is to write down all integers from 1 to N, then to leave 2 and strike out all even numbers after 2, then leave 3 and strike out every third integer after 3, next leave 5 and strike out every fifth integer, and so on. What remain are the primes, and if the procedure is followed up to N, then the nature of N will be revealed. This is obviously a lengthy process. However, the computation can be shortened by generating only the primes up to $\sqrt{N}$ and testing these as divisors of N. As you test N with integers below $\sqrt{N}$, you are also "automatically" eliminating the possible divisors *above* $\sqrt{N}$. The possible divisors converge at $\sqrt{N}$.

The following program makes it easy to reduce the number of integers eliminated and to perform division testing on a larger set of numbers. For example, multiples of 2 and 3 could be eliminated, as is done in the program for prime factorization, but it is computationally simpler and still efficient to eliminate the even divisors only. The algorithm for testing N is to divide N by all odd numbers greater than 2 and to terminate the procedure when a divisor is found or if the testing divisor is greater than $\sqrt{N}$.

The length of time the test requires depends on how many times the program loops through subroutine [B] . Be sure to allow several seconds for testing large numbers.

1

Prime Number Testing



The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [A] [STO] 1 [x:t] 1 [2nd] [x=t] [E] 3 [2nd] [x=t] [C] [RCL] 1 [÷] 2 [=] [INV] [2nd] [Int] [x:t] 0 [2nd] [x=t] [D] 3 [STO] 2 [RCL] 1 [√x] [STO] 03	Stores N and checks to see if $N \leq 1$. If $N \leq 1$, program transfers to [E]. Divides N by 2 and checks for remainder. If remainder = 0, transfers to [D] Checks to see if $N \leq 3$; if so, transfers to [D]. Stores 3 and $\sqrt{N}$ for later testing.
030	[2nd] [Lbl] [B] [RCL] 1 [÷] [RCL] 2 [=] [INV] [2nd] [Int] [x:t] 0 [2nd] [x=t] [D] 2 [SUM] 2 [RCL] 2 [x:t] [RCL] 3 [2nd] [x=t] [B]	Divides N by 3 and checks for remainder. If remainder = 0, transfers to [D]. If remainder $\neq 0$, increases divisor by 2 and tests against $\sqrt{N}$. If divisor $< \sqrt{N}$, loops back to [B].
054	[2nd] [Lbl] [C] [RCL] 1 [2nd] [Prt] [R/S]	N is prime. Recalls N and stops.
060	[2nd] [Lbl] [D] [RCL] 1 [x^2] [+/−] [√x] [R/S]	N is not prime. Recalls flashing N and stops.
068	[2nd] [Lbl] [E] 0 [R/S]	$N \leq 1$ (not prime by definition). Clears display and stops.
	[LRN] [RST]	Exits learn mode.

Note: Print command may be omitted if the PC-100A is not used.

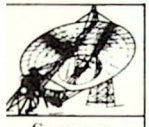
Register Contents

R_{00}
R_{01} N
R_{02} Used
R_{03} Used
R_{04}

USING THE PROGRAM

Procedure	Press	Display/Comments
Enter N.	A	N, if N is prime.* Flashing N, if N is not prime. 0, if N≤1.

*This value is printed if the PC-100A is used.



SAMPLE PROBLEM

Determine if 8719 is prime.

Procedure	Press	Display/Comments
Enter program. Enter 8719.	8719 A	8719. (in approximately 47 seconds)

8719 is a prime.

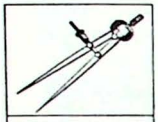
FURTHER EXPLORATIONS

1. Determine if 9161, 7351, 5559, and 4513 are prime. (*Answer:* Only 5559 is not prime.)
2. Find the smallest integer N such that $2^{2^{N+1}} - 1$ is not a prime. (*Answer:* N = 5)
3. Rewrite the program in this section so that all multiples of 2 and 3 are stricken out before testing divisors (see the program for prime factorization).

REFERENCES/ INFORMATION

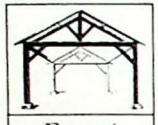
C. T. Long, *Elementary Introduction to Number Theory* (Lexington, Mass.: D. C. Heath and Company, 1972), pp. 59-73.

1 Prime Factorization



OBJECTIVE

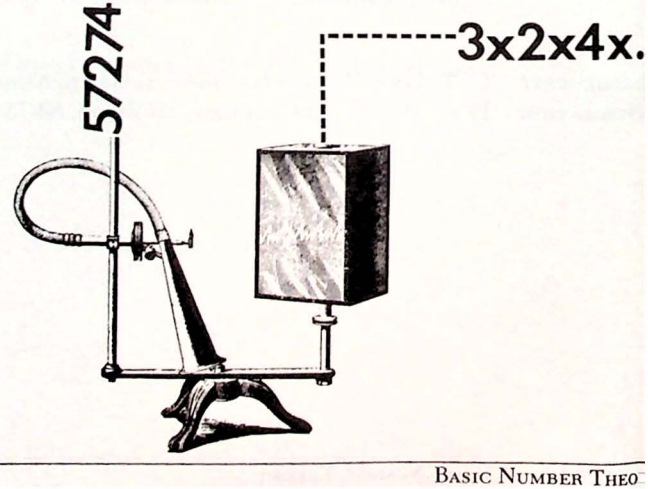
The Fundamental Theorem of Algebra tells us that each positive integer can be factored into a product of primes. This information is useful in finding common divisors of numbers, in factoring polynomials, and in simplifying the roots of numbers.

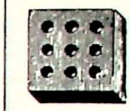


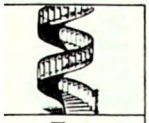
FACTS/ THEORY

It is obvious that the prime factors for a number N can be determined by dividing N by the integers less than N. This is not the most efficient method for computing the factors, however, since division by non-primes less than N yields no information. A better procedure would be to test only the primes less than N, but generation of the primes within a computational algorithm is time consuming. The algorithm used in this program is a combination of the above methods.

Given a number N, divide by 2,3, and all larger odd integers (5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, . . .). Although there are non-primes in this sequence which will provide no information, this sequence is easier to generate than only primes. Division is done successively by numbers in the sequence, and as each factor x is found, N is replaced by N/x before dividing by x again to see if it is a square factor or testing the next number in the sequence. Note that, if this procedure is followed, it is only necessary to test numbers up to the square root of N, since no number larger than $\sqrt{N}$ can be a square factor, and non-square factors will be divided out by numbers less than $\sqrt{N}$.







THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A INV 2nd St flg 1 INV 2nd St flg 2 STO 05	Stores N and insures that flags 1 and 2 are not set.
010	2nd Lbl B 2nd B' RCL 5 $\sqrt{x}$ x:t RCL 1 2nd x=t 2nd A' 0 STO 06	Transfers to subroutine B' to generate divisors (x); then tests to see if $x \geq \sqrt{N}$. Clears exponent counter (memory 6).
024	2nd Lbl C RCL 5 ÷ RCL 1 = STO 4 INV 2nd Int x:t 0 2nd x=t D RCL 6 x:t 0 2nd x=t B RCL 1 2nd Prt R/S RCL 6 2nd Prt 2nd Adv R/S GTO B	Divides N by generated sequence of divisors, testing each for divisibility.
057	2nd Lbl D 2nd Op 26 RCL 4 STO 5 x:t 1 INV 2nd x=t C RCL 1 2nd Prt R/S RCL 6 2nd Prt 2nd Adv R/S	Increments exponent counter by 1. Replaces N by N/x and checks to see if N/x = 1.
079	2nd Lbl E RCL 1 x^2 +/− $\sqrt{x}$ R/S	Display flashes and program stops, signifying that all factors have been found.
087	2nd Lbl 2nd A' RCL 5 2nd Prt R/S 1 2nd Prt 2nd Adv R/S E	Displays last x.
098	2nd Lbl 2nd B' INV 2nd If flg 1 2nd D' INV 2nd If flg 2 2nd E' 2 SUM 1 INV SBR	Tests for divisors 2 and 3; increases x by 2 to generate sequence of divisors.
112	2nd Lbl 2nd D' 2 STO 1 2nd St flg 1 INV SBR	Stores 2 as first divisor and sets flag 1.
120	2nd Lbl 2nd E' 3 STO 1 2nd St flg 2 INV SBR	Stores 3 as divisor and sets flag 2.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

1

Prime Factorization

Register Contents

R_{00}		R_{05}	N; factor
R_{01}	Used	R_{06}	exponent counter
R_{02}		R_{07}	
R_{03}		R_{08}	
R_{04}	Used	R_{09}	

Using the Program

Procedure	Press	Display/Comments
Enter N.	A	$factor_1$°
	R/S	$exponent_1$°
	R/S	$factor_2$°
	R/S	$exponent_2$°
	. . .	. . .
	R/S	$factor_n$° (last prime
	R/S	$exponent_n$° factor)
	R/S	flashing display to show that last prime factor has been found.

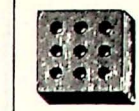
°This value is printed if the PC-100A is used.

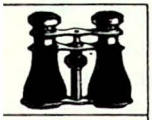
Sample Problem

Find the prime factors of 707940.

Procedure	Press	Display/Comments
Enter program.		**0**
Enter value of N and calculate prime factors.	707940 A	**2.** $Factor_1$ } 2^2
	R/S	**2.** $Exponent_1$ }
	R/S	**3.** $Factor_2$ } 3^4
	R/S	**4.** $Exponent_2$ }
	R/S	**5.** $Factor_3$ } 5^1
	R/S	**1.** $Exponent_3$ }
	R/S	**19.** $Factor_4$ } 19^1
	R/S	**1.** $Exponent_4$ }
	R/S	**23.** $Factor_5$ } 23^1
	R/S	**1.** $Exponent_5$ }
	R/S	**21.** flashing — end of run.

The prime factors of 707940 are $(2)^2$, $(3)^4$, $(5)^1$, $(19)^1$, and $(23)^1$.



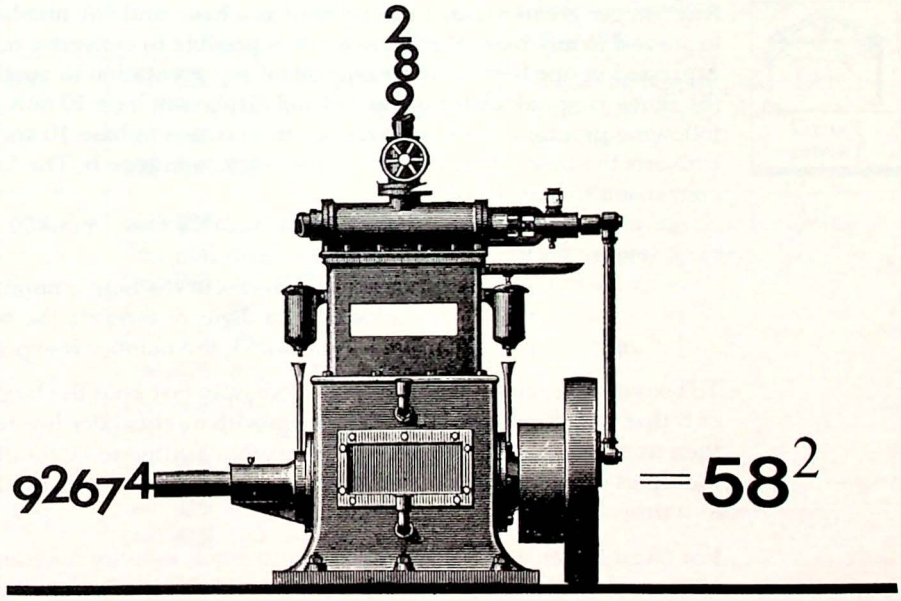


FURTHER EXPLORATIONS

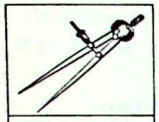
1. Determine the prime factors of
   (a) 7286. (*Answer:* 2^1 and 3643^1)
   (b) 94860. (*Answer:* 2^2, 3^2, 5^1, 17^1, and 31^1)
2. A perfect number is one where the sum of its divisors is equal to two times the number; for example, the divisors of 6 are 1, 2, 3, and 6, and $1+2+3+6=12$ (or 2×6). Is 8128 a perfect number?
3. There is a theorem: An even integer is perfect if and only if it has the form $2^{P-1}(2^P-1)$, where 2^P-1 is a prime. Check this theorem for P = 2, 7, 13, and 19.

REFERENCES/ INFORMATION

C. T. Long, *Elementary Introduction to Number Theory* (Lexington, Mass.: D. C. Heath and Company, 1972), pp. 42-47, 59-61.



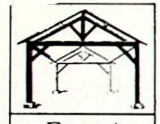
1 Base Conversion



OBJECTIVE

Numbers exist independent of the names or symbols we use to describe them. For example, consider the number 1341, which we call one thousand, three hundred, and forty-one. Because the Hindu-Arabic number system commonly used today is based on 10, the number 1341 means $1+(4\times10)+(3\times10^2)+(1\times10^3)$. In other words, the number is interpreted as a sum of multiples of powers of 10.

Bases other than 10 can also be used, and in this case the symbol 1341 would be interpreted differently. In base 5 it would represent $1+(4\times5)+(3\times5^2)+(1\times5^3)$. Using different bases to interpret numbers is both an interesting and a useful exercise. For instance, number representation in base 2 has proved to be extremely important in computer-related activities, since a computer typically stores and manipulates numbers in devices which only have two states — "ON and OFF," which can be used to represent 1 and 0.



FACTS/ THEORY

Any integer greater than 1 can be used as a base, and any number can be expressed in any base. Furthermore, it is possible to convert a number expressed in one base (a) to its equivalent representation in another base (b). Since your calculator computes and displays in base 10 only, the following program makes conversions from base a to base 10 and then converts the base 10 result to its representation in base b. The formula for conversion to base 10 is:

$$N=(x_n a^{n-1})+(x_{n-1}a^{n-2})+(x_{n-2}a^{n-3})+\ \ldots\ +(x_1a^0)$$

where N = the *base 10* expression,
x_i = the digits or terms of the *base a* number,
n = the number of digits or terms in the *base a* number
and a = the base in which the number is expressed.

To convert from base 10 to base b, the program first finds the largest power of b that will divide the base 10 number with a remainder less than b. It then uses an iterative division/multiplication routine to factor out the multiples of powers of b, generating the base b number one digit or term at a time.

For bases larger than 10, the additional symbols used for notation are the integers themselves (i.e., 10, 11, 12, 13, etc.) because the calculator will display only numeric values. These are presented as multi-digit single terms in the program, and, when converting a number in base a that contains a multi-digit term, these are entered in subroutine [B] as single terms (see example 2).

As written, the entire program operates only on positive whole numbers, although subroutine [B] can be used to convert decimals in base a to decimals in base 10 by entering a 0 as the number of terms.

THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [A] [2nd] [CMs] [STO] 1 [R/S] [STO] 2 [R/S] [−] 1 [=] [STO] 5 [R/S]	Clears memories and stores base a, base b, and number of terms, less 1, in the appropriate memories.
015	[2nd] [Lbl] [B] [×] [(] [RCL] 1 [y^x] [RCL] 5 [)] [EE] [INV] [EE] [=] [SUM] 00 1 [+/−] [SUM] 5 [R/S]	Converts base a to base 10. (Base 10 number stored in memory 0.) *Note:* the [EE] [INV] [EE] sequence truncates the guard digits and insures proper rounding.
036	[2nd] [Lbl] [D] [RCL] 2 [$x \rightleftarrows t$] 1 [SUM] 4 [RCL] 0 [÷] [RCL] 2 [y^x] [RCL] 4 [=] [INV] [2nd] [$x \rightleftarrows t$] [2nd] [A'] [D]	Finds the largest power of base b that will divide the base 10 number.
057	[2nd] [Lbl] [2nd] [A'] [RCL] 0 [÷] [RCL] 2 [y^x] [RCL] 4 [=] [2nd] [Int] [2nd] [Prt] [R/S] [×] [(] [RCL] 2 [y^x] [RCL] 4 [)] [EE] [INV] [EE] [=] [INV] [SUM] 0 [RCL] 4 [−] 1 [=] [STO] 04 0 [$x \rightleftarrows t$] [RCL] 4 [2nd] [$x \rightleftarrows t$] [2nd] [B'] [2nd] [A']	Generates the base b number one term at a time.
100	[2nd] [Lbl] [2nd] [B'] [RCL] 0 [2nd] [Prt] [R/S]	Finds final term and stops.
	[LRN] [RST]	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

1

Base Conversion

Register Contents

R_{00}	Used	R_{05}	Used
R_{01}	a	R_{06}	
R_{02}	b	R_{07}	
R_{03}		R_{08}	
R_{04}	Used	R_{09}	

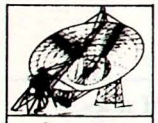
Using the Program

Procedure	Press	Display/Comments
Enter base a.	A	a
Enter base b.	R/S	b
Enter number of terms in N_a.	R/S	(n-1)
Enter each term in N_a. x_1	B	-1.
x_2	B	-1.
. . .	. . .	. . .
x_n	B	-1.

Note: When each term has been entered, N_a has been converted to N_{10}. To see N_{10}, recall memory 0.

Now, convert to base b.	D	x_1°	First term.
	R/S	x_2°	Second term.
	R/S	x_3°	Third term.
	. . .	. . .	
	R/S	x_n°	Final term.
	R/S	flashing 0	Signifies last term has been generate

°This value is printed if the PC-100A is used.



Sample Problem

1. Express 2916 in base 7.

Procedure	Press	Display/Comments
Enter program.		0
Enter a (10).	10 A	10.
Enter b (7).	7 R/S	7.
Enter number of terms in N.	4 R/S	3.

(When converting base 10 to another base, you can bypass subroutine B by storing N directly in memory 0 and going on to subroutine D.)

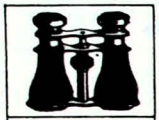
Procedure	Press	Display/Comments
Enter N (2916).	2916 STO 00	2916.
Convert to base b (7).	D	1. x_1
	R/S	1. x_2
	R/S	3. x_3
	R/S	3. x_4
	R/S	4. x_5
	R/S	flashing 0. End of run.

2916 in base 10 equals 11334 in base 7.

2. Express $[84(13)2]_{15}$ in base 11. (Note that 13 in this number is a single term; therefore, the number contains four terms.)

Procedure	Press	Display/Comments
Enter base a (15).	15 A	15.
Enter base b (11).	11 R/S	11.
Enter number of terms in N (4).	4 R/S	3.
Convert to base 10, one term at a time.	8 B	-1. x_1
	4 B	-1. x_2
	13 B	-1. x_3
	2 B	-1. x_4
Convert to base b (11).	D	1. x_1
	R/S	10. x_2
	R/S	1. x_3
	R/S	2. x_4
	R/S	3. x_5
	R/S	0. flashing — end of run.

[84(13)2] in base 15 equals [1(10)123] in base 11.

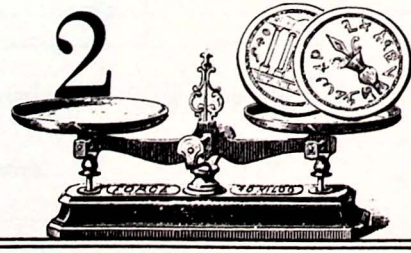


FURTHER EXPLORATIONS

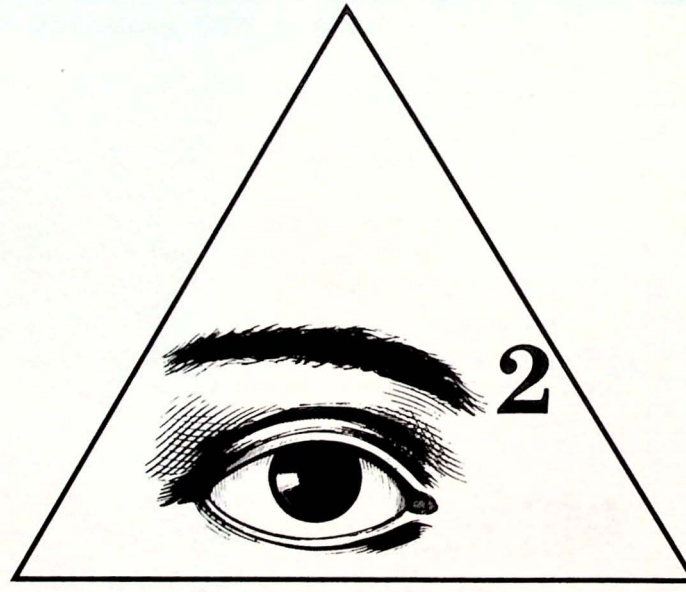
1. What numbers in base 10 are represented by $[2(11)41]_{16}$ and $(7)_8$? *(Answers:* 11073 and 7)
2. Revise the program to perform conversion of decimals in base a to decimals in base b.
3. Modify the program to perform conversion of negative integers from base a to base b.
4. Construct addition and multiplication tables for base 4 and base 8.

REFERENCES/ INFORMATION

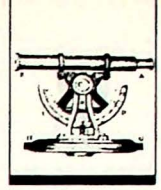
C. T. Long, *Elementary Introduction to Number Theory* (Lexington, Mass.: D. C. Heath and Company, 1972), pp. 24-29.



2 College Algebra and Trigonometry



Introduction



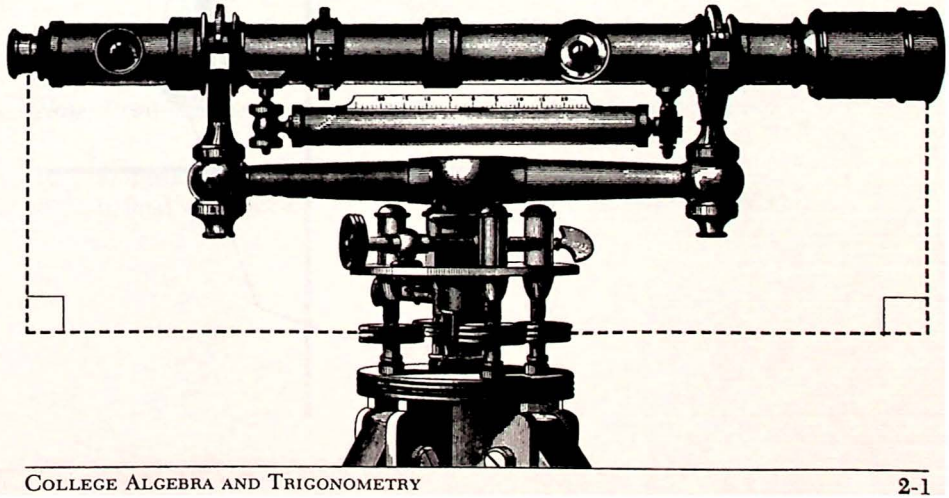
After basic arithmetic, algebra and trigonometry are probably the mathematical tools we use most extensively. They are usually considered to be the foundations upon which we build a wide variety of powerful mathematical applications in such areas as calculus, analytical geometry, physics, chemistry, biology — any field that requires exploration with mathematics.

Algebra deals with general statements (equations or formulas) of mathematical relationships, while trigonometry applies specifically to the relationships between the sides and angles of triangles. Your calculator's special algebraic and trigonometric function keys — and its programmability — can help you greatly in both these areas. Because computations of this type usually require much repetitive calculation, they are ideally suited to the programmable calculator.

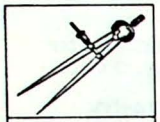
This chapter illustrates some of the applications of your calculator in algebra and trigonometry. Included are these topics:

- evaluating and graphing algebraic functions,
- evaluating polynomial expressions,
- solving linear and quadratic equations,
- computing the area, sides, and angles of triangles, and
- graphing trigonometric functions.

Although this is certainly not an exhaustive list of applications, we hope these examples will open the door and help you find new ways to use your calculator with algebra and trigonometry.



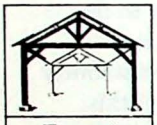
2 Graphing Functions



OBJECTIVE

"A picture is worth a thousand words." Many times a picture can explain some difficult situation or illustrate relationships and functional behavior much better than any verbal description ever could.

The following program calculates ordered pairs of a function f(x) over an interval from x = a to x = b with increments of d. You can then easily plot the ordered pairs to get a picture of the function.



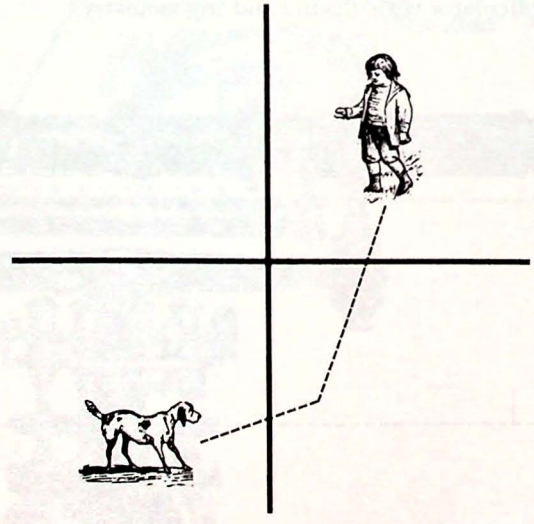
FACTS/ THEORY

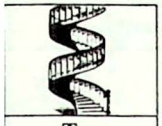
Typically a function y = f(x) may be graphically "pictured" on a rectangular coordinate system as a set of points. Each point is represented by a pair of numbers (x,f(x)) where for each x there corresponds some value f(x) which is determined by the function. Usually x takes on values from some initial value, a, to some final value, b, with a fixed increment or step of d. For example, the graph of the function

$$f(x) = x^2 + 2x + 1$$

from x = 1 to x = 4 and with d = 0.5 would consist of the points (1,f(1)), (1.5,f(1.5)), (2,f(2)), (2.5,f(2.5)), (3,f(3)), (3.5,f(3.5)), (4,f(4)).

In order to use the following program you need to be able to enter the function f(x) into the program as a subroutine. This is easy to do if you follow the step by step procedure outlined in the "program" section, and illustrated in the example. (For extra information on subroutines and their use as part of programs see *Personal Programming*, pages IV-46 ff.)





THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [A] [STO] 1 [R/S]	Enters a and stores it in memory 01.
005	[2nd] [Lbl] [B] [STO] 2 [R/S]	Enters b and stores it in memory 02.
010	[2nd] [Lbl] [C] [STO] 3 [R/S]	Enters d and stores it in memory 03.
015	[2nd] [Lbl] [D] [RCL] 1 [2nd] [Prt] [R/S]	Starts the program run and displays x.
021	[E] [STO] 4 [2nd] [Prt] [2nd] [Adv] [R/S]	Transfers control to subroutine [E] where f(x) is calculated. Displays f(x).
027	[RCL] 3 [SUM] 1 [RCL] 1 [x:t] [RCL] 2 [2nd] [x≥t] [D] [CLR] [R/S]	Increments x by d and checks to see if $x \geq b$. If $x \leq b$, f(x) is calculated. If $x > b$, display is cleared and program stops.
040	[2nd] [Lbl] [E] You insert your function keystrokes here, with memory 01 used as the value for the x variable.	Labels the subroutine where f(x) will be calculated.
	[LRN] [RST]	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

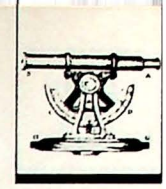
REGISTER CONTENTS

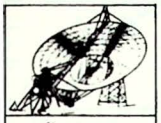
R_{01} a, initial value for x
R_{02} b, final value for x
R_{03} d, step value
R_{04} f(x) (current value of x)

USING THE PROGRAM

Procedure	Press	Display/Comments
Transfer the program to subroutine [E].	[GTO] [E]	0
Place the calculator in the learn mode and program the steps for finding f(x). (Use [RCL] **01** whenever you need the value of the x variable.)	[LRN] Enter your program for f(x).	043 00
Make sure the last step is [INV] [SBR].	[INV] [SBR]	(This step returns the calculator back to the mair body of the program.)
Take the calculator out of the learn mode of operation.	[LRN]	
Enter the initial x value, a (where you'd like the graph to begin).	[A]	a
Enter the final x value, b.	[B]	b
Enter the step value, or increment, d.	[C]	d
Start the program.	[D]	a°
	[R/S]	f(a)°
Continue to press [R/S] for each new x and f(x).	...	...
	[R/S]	b°
	[R/S]	f(b)°
	[R/S]	0 Signifies end of run.

°This value is printed if the PC-100A is used.





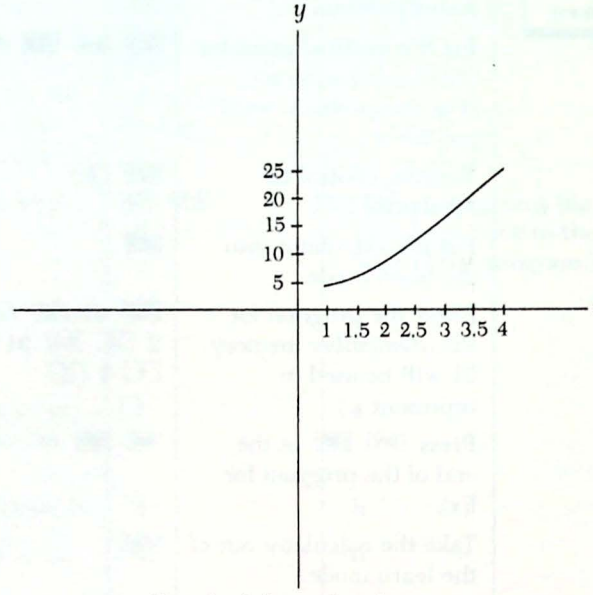
SAMPLE PROBLEM

Let's use the program to find ordered pairs which will allow us to graph $f(x) = x^2 + 2x + 1$ from $x = 1$ to $x = 4$ with increments of 0.5.

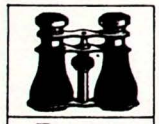
Procedure	Press	Display/Comments
Enter program.		0
Fix the decimal point for the desired accuracy. (For this problem we'll use 2.)	CLR 2nd fix 2	0.00
Transfer control to subroutine E.	GTO E	0.00
Put the calculator into the learn mode.	LRN	042 00
Enter the program for f(x).(*Remember* memory 01 will be used to represent x.)	RCL 01 x^2 + 2 × RCL 01 + 1 =	
Press INV SBR at the end of the program for f(x).	INV SBR	
Take the calculator out of the learn mode.	LRN	0.00
Enter a.	1 A	1.00
Enter b.	4 B	4.00
Enter d.	.5 C	0.50
Start the program.	D	1.00 a
Continue to press R/S to find each new x and f(x).	R/S	4.00 f(a)
	...	...
	R/S	4.00 b
	R/S	25.00 f(b)
	R/S	0 End of run.

Values

x	y
1	4
1.5	6.25
2	9
2.5	12.25
3	16
3.5	20.25
4	25



Sketch of the ordered pairs



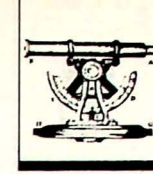
FURTHER EXPLORATIONS

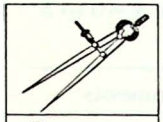
1. Graph $f(x) = 5x^3 - 2x + 4$ for $x = 0$ to $x = 1$ with increments of 0.1
2. Graph $f(x) = 2x^2 - \frac{x}{2} + \frac{1}{3}$ for $x = 10$ to $x = 20$ with increments of 2.
3. Find the roots of $f(x) = x^2 - 4x - 1.5$. The roots of the equation are those points, x, where $f(x) = 0$. (*Answers:* -0.345 and 4.345.)
4. Does the function $f(x) = 4x^3 - 3x^2 + 2x + 1$ have any maximum points in the range $x = -10$ to $x = 0$? Caution: What happens when **2** [+/-] [y^x] **3** [=] is pressed? How can you handle raising a negative number to a power?
5. What limit does the function $f(x) = \frac{10}{x^2 + 1}$ approach as x gets larger and larger?

REFERENCES/ INFORMATION

See *Personal Programming* — Subroutines, pp. IV-48 ff.

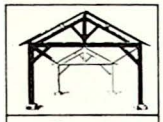
Polynomial Evaluations





OBJECTIVE

Many problems in economics, physics, chemistry, business, and finance, as well as in mathematics, require the evaluation of polynomial expressions. The Master Library module has a program (ML-07) that will evaluate any polynomial expression for any real number. This can be useful when graphing a polynomial function is desired.



FACTS/ THEORY

Generally, a polynomial function, P(x), is written in the following way:

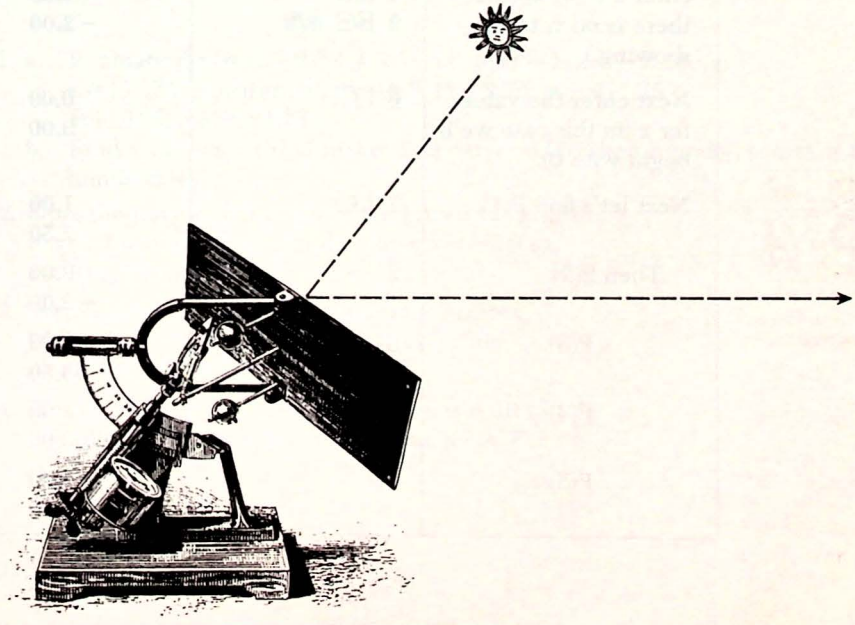
$$P(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$$

where a_i is the coefficient of the variable x^i for $i = 0,1,2, \cdots, n$.

Program ML-07 has a maximum value of n which is limited by the number of data registers available. The number of registers available must be equal to or greater than the value $n+6$. Normally in the TI Programmable 58 the calculator can handle up to and including $n=24$, while the TI Programmable 59 can handle up to and including $n=54$.

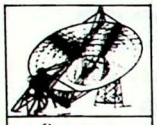
These limits can be increased by using the repartitioning method outlined in program ML-02 (p. 9). Further help on repartitioning may be found in *Personal Programming*, pages V-22,29.

The Master Library manual has additional information on using the program, so we'll go directly to our sample problem.



2

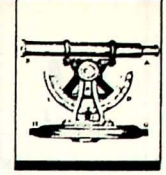
Polynomial Evaluations



Sample Problem

Let's use the program to help us graph $P(x) = 3 - 2.5x + 4x^3 - 2x^4$ for $x = 0$ to 5 with increments of 1.

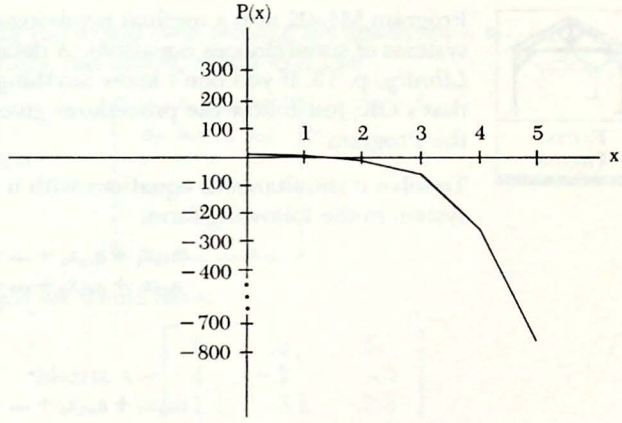
Procedure	Press	Display/Comments
Select the program from the Master Library.	CLR 2nd Pgm 07	0.
Fix the decimal to the desired accuracy. (Here we'll use two places.)	2nd Fix 2	0.00
Enter n, the highest power of x.	4 A	4.00 n
Now enter a 0 to tell the calculator we'll enter a_0 first.	0 B	0.00
Enter each coefficient starting with a_0. (Notice that we must enter a 0 for a_2 since there is no x_2 term showing.)	3 R/S 2.5 +/− R/S 0 R/S 4 R/S 2 +/− R/S	3.00 a_0 −2.50 a_1 0.00 a_2 4.00 a_3 −2.00 a_4
Next enter the value for x (in this case we'll begin with 0).	0 C	0.00 x = 0.00 3.00 P(x) = 3.00
Next let's find P(1).	1 C	1.00 x = 1.00 2.50 P(x) = 2.50
Then P(2)	2 C	2.00 x = 2.00 −2.00 P(x) = −2.00
P(3)	3 C	3.00 x = 3.00 −58.50 P(x) = −58.5
P(4)	4 C	4.00 x = 4.00 −263.00 P(x) = −263
P(5)	5 C	5.00 x = 5.00 −759.50 P(x) = −759.5

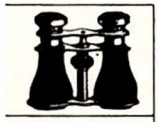


The graph of the polynomial from $x = 0$ to $x = 5$ with increments of 1 would look like the following:

Table

x	$P(x)$
0	3
1	2.5
2	−2
3	−58.5
4	−263
5	−759.5





FURTHER EXPLORATIONS

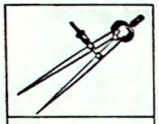
1. a. Evaluate $P(x) = 12 - 2x^2 + 4.5x^3 + 2x^5$ for $x = 1$, $x = -2.75$, and $x = 11.175$. (*Answer:* $(P(1) = 16.5, P(-2.75) = -411.26, P(11.175) = 354594.23)$
   b. Find a value for x that makes $P(x) = 0$ (or 0.00 when rounded to nearest hundredths).
2. Does the polynomial $P(x) = -1 + 4x^2 - 2.1x^3 + 3.2x^4$ have a maximum value in the range $x = -10$ to $x = 10$? (*Answer:* No.)
3. Sketch graphs of the power functions $P(x) = x$, $P(x) = x^2$, $P(x) = x^3$, $P(x) = x^4$, and $P(x) = x^5$ for $x = -10$ to $x = 10$.
4. Sketch the graph of $P(x) = x^3 - x^2$ for $x = -10$ to $x = 10$. Is there a maximum in the range? A minimum?

REFERENCES/ INFORMATION

Master Library, pp. 24-26.

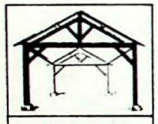
Personal Programming, pp. V-22, V-29.

2 Systems of Linear Equations (Master Library Program)



OBJECTIVE

Solving simultaneous equations can be a time-consuming process — and one often prone to simple arithmetic errors. However, there is a program in the Master Library that can take the drudgery out of solving simultaneous equations. Program ML-02 can normally solve up to six equations with six unknowns using the TI Programmable 58 and up to eight equations with eight unknowns using the TI Programmable 59.



FACTS/ THEORY

Program ML-02 uses a method involving matrices and determinants to solve systems of simultaneous equations. A detailed explanation is given in *Master Library*, p. 13. If you don't know anything about matrices and determinants, that's OK; just follow the procedures given below and in the section "Using the Program."

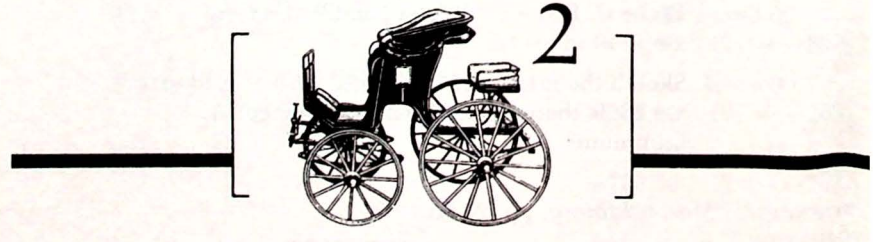
To solve n simultaneous equations with n unknowns, you must first write the system in the following form:

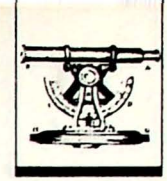
$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

where for $i = 1, 2, 3, \ldots, n$ and $j = 1, 2, 3, \ldots, n$; a_{ij} and b_i are fixed values and x_j is the unknown.

Suppose we have the following set of equations:

$$\begin{aligned} 2x + 3y + 5z &= 12 \\ 4x - 2y + 3.5z &= 7.5 \\ 5.2y - 1.1x - 6.3z &= -24 \end{aligned}$$





To solve the set of equations simultaneously we would rewrite the set of equations as:

$$2x_1 + 3x_2 + 5x_3 = 12$$
$$4x_1 - 2x_2 + 3.5x_3 = 7.5$$
$$-1.1x_1 + 5.2x_2 - 6.3x_3 = -24$$

where $x_1 = x$, $x_2 = y$, and $x_3 = z$.

To use program ML-02, you would then arrange the coefficients, a_{11}, a_{12}, ..., a_{nn}, in a matrix. Generally, this matrix would look like

$$\text{Matrix A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

For our specific example we would have:

$$\text{Matrix A} = \begin{bmatrix} 2 & 3 & 5 \\ 4 & -2 & 3.5 \\ -1.1 & 5.2 & -6.3 \end{bmatrix}$$

Note that any a_{ij} may be zero for a particular i and j but it must still be entered in the appropriate place in the matrix.

The program calculates what is called the determinant for matrix A and uses the determinant in solving the system for a set of values, x_1, x_2, ..., x_n, that will make each equation in the set a true equation. If the determinant is zero, there is no solution for the particular set of equations you are trying to solve.

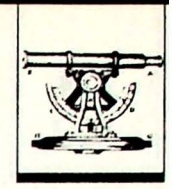
2

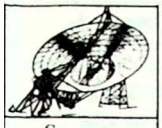
Systems of Linear Equations (Master Library Program)

Using the Program

Procedure	Press	Display/Comments
Select the program from the Master Library.	CLR 2nd Pgm 02	0.
Enter the number of unknowns, n.	A	n°
Enter the column number (j) you are going to begin entering. This is usually 1.	B	j
Enter each coefficient starting with the first column:		
a_{11}	R/S	a_{11}°
a_{21}	R/S	a_{21}°
•••	•••	•••
a_{nn}	R/S	a_{nn}°
(*Note:* To correct an entry in column j, press j B . Then reenter the entire column using the R/S key.)		
Calculate the determinant used to find each unknown (you don't need to know what this means).	C	determinant°
Enter a 1 to begin the process of entering the b values.	1 D	1
Enter each b_1:		
b_1	R/S	b_1°
•••	•••	•••
b_n	R/S	b_n°
Calculate each x. Start by entering 1 to find x_1; then press R/S for each x_1.	CLR E	1.
	1 2nd A'	1.
	R/S	x_1°
	R/S	x_2°
	•••	•••

°This value is printed if the PC-100A is used.





SAMPLE PROBLEM

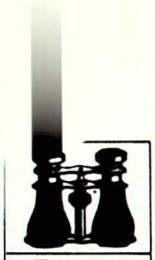
Let's use the program to solve the system of simultaneous equations in our example:

$$2x_1 + 3x_2 + 5x_3 = 12$$
$$4x_1 - 2x_2 + 3.5x_3 = 7.5$$
$$-1.1x_1 + 5.2x_2 - 6.3x_3 = -24$$

Procedure	Press	Display/ Comments	
Fix the decimal point for the desired accuracy (in this problem we'll use 3).	CLR 2nd Fix 3	0.000	
Select the program.	2nd Pgm 02	0.000	
Enter the number of unknowns (here n = 3).	3 A	3.000	
Enter starting column number (j).	1 B	1.000	
Enter each a_{ij} starting with the first column. (Enter each a_{ij} even if a_{ij} is 0.)	2 R/S	2.000	
	4 R/S	4.000	
	1.1 +/- R/S	-1.100	
	3 R/S	3.000	
	2 +/- R/S	-2.000	
	5.2 R/S	5.200	
	5 R/S	5.000	
	3.5 R/S	3.500	
	6.3 +/- R/S	-6.300	
Calculate the determinant.	C	145.850	It will take about 10 - 20 seconds to find the value.
Enter in a 1 to begin the process of entering the b values.	1 D	1.000	
Enter each b_i.	12 R/S	12.000	
	7.5 R/S	7.500	
	24 +/- R/S	-24.000	

Systems of Linear Equations (Master Library Program)

Procedure	Press	Display/Comments
Now we're ready to calculate each x. First begin the routine.	CLR E	1.000 Notice it takes 10-15 seconds to get the display.
Then enter 1 to begin to find each x.	1 2nd A	1.000
Press R/S to find each x.	R/S	−1.525 This is x_1.
	R/S	−0.748 This is x_2.
	R/S	3.459 This is x_3.



Further Explorations

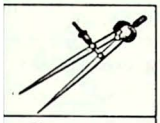
Solve the following systems of simultaneous equations:

a. $2x_1 - 3.5x_2 + 6x_3 = -1$
$4x_1 + 6x_2 - 3x_3 = 21$
$12x_1 - 3x_2 + 4x_3 = 17$

b. $3x_1 + 4x_3 - 7x_4 = 2.7$
$x_1 + 3x_2 - 7x_3 + 12x_4 = 11.7$
$4x_1 - 2x_2 + 6x_3 + 7x_4 = 21$
$5x_1 + 6x_2 + 9x_4 = 13$

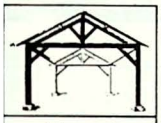
References/ Information *Master Library,* pp. 9-13.

Quadratic Equations



OBJECTIVE

One of the most practical formulas found in algebra is the quadratic formula used to find the roots of a quadratic equation. The following program can be used to find the roots of any quadratic equation, regardless of whether the roots are real or complex.



FACTS/ THEORY

The general form of a quadratic equation is

$$ax^2 + bx + c = 0$$

Where $a \neq 0$. The quadratic formula used to find the roots of the equation is expressed as

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

or

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \text{ and } x_2 = \frac{-b - \sqrt{b_2 - 4ac}}{2a}$$

where x_1 and x_2 are the roots.

The key to whether the roots are real or complex is found in that part of the formula called the *discriminant*,

$$\sqrt{b^2 - 4ac}$$

If $b^2 - 4ac > 0$, both roots are real.
If $b^2 - 4ac = 0$, both roots are real and the same.
If $b^2 - 4ac < 0$, both roots are complex.

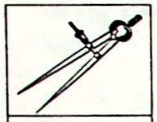
When the roots are complex, they can be divided into their real and imaginary parts as follows:

$$x_1 = R + id \text{ and } x_2 = R - id$$

where $R = -b/2a$,
$i = \sqrt{-1}$, and
$d = \dfrac{\sqrt{4ac - b^2}}{2a}$

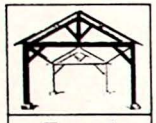
The following program displays both roots if the roots are real and the values for R and d if the roots are complex.

2 Systems of Linear Equations (Master Library Program)



OBJECTIVE

Solving simultaneous equations can be a time-consuming process — and one often prone to simple arithmetic errors. However, there is a program in the Master Library that can take the drudgery out of solving simultaneous equations. Program ML-02 can normally solve up to six equations with six unknowns using the TI Programmable 58 and up to eight equations with eight unknowns using the TI Programmable 59.



FACTS/ THEORY

Program ML-02 uses a method involving matrices and determinants to solve systems of simultaneous equations. A detailed explanation is given in *Master Library*, p. 13. If you don't know anything about matrices and determinants, that's OK; just follow the procedures given below and in the section "Using the Program."

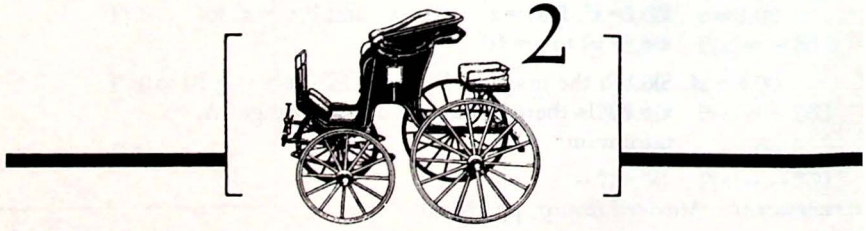
To solve n simultaneous equations with n unknowns, you must first write the system in the following form:

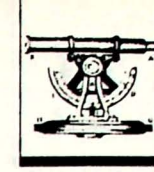
$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

where for $i = 1, 2, 3, \ldots, n$ and $j = 1, 2, 3, \ldots, n$; a_{ij} and b_i are fixed values and x_j is the unknown.

Suppose we have the following set of equations:

$$\begin{aligned} 2x + 3y + 5z &= 12 \\ 4x - 2y + 3.5z &= 7.5 \\ 5.2y - 1.1x - 6.3z &= -24 \end{aligned}$$





To solve the set of equations simultaneously we would rewrite the set of equations as:

$$2x_1 + 3x_2 + 5x_3 = 12$$
$$4x_1 - 2x_2 + 3.5x_3 = 7.5$$
$$-1.1x_1 + 5.2x_2 - 6.3x_3 = -24$$

where $x_1 = x$, $x_2 = y$, and $x_3 = z$.

To use program ML-02, you would then arrange the coefficients, a_{11}, a_{12}, ..., a_{nn}, in a matrix. Generally, this matrix would look like

$$\text{Matrix A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

For our specific example we would have:

$$\text{Matrix A} = \begin{bmatrix} 2 & 3 & 5 \\ 4 & -2 & 3.5 \\ -1.1 & 5.2 & -6.3 \end{bmatrix}$$

Note that any a_{ij} may be zero for a particular i and j but it must still be entered in the appropriate place in the matrix.

The program calculates what is called the determinant for matrix A and uses the determinant in solving the system for a set of values, x_1, x_2, ..., x_n, that will make each equation in the set a true equation. If the determinant is zero, there is no solution for the particular set of equations you are trying to solve.

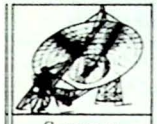
2

Systems of Linear Equations (Master Library Program)

Using the Program

Procedure	Press	Display/Comments
Select the program from the Master Library.	CLR 2nd Pgm 02	0.
Enter the number of unknowns, n.	A	n°
Enter the column number (j) you are going to begin entering. This is usually 1.	B	j
Enter each coefficient starting with the first column:		
a_{11}	R/S	a_{11}°
a_{21}	R/S	a_{21}°
•••	•••	•••
a_{nn}	R/S	a_{nn}°
(*Note:* To correct an entry in column j, press j B . Then reenter the entire column using the R/S key.)		
Calculate the determinant used to find each unknown (you don't need to know what this means).	C	determinant°
Enter a 1 to begin the process of entering the b values.	1 D	1
Enter each b_1:		
b_1	R/S	b_1°
•••	•••	•••
b_n	R/S	b_n°
Calculate each x. Start by entering 1 to find x_1; then press R/S for each x_1.	CLR E	1.
	1 2nd A'	1.
	R/S	x_1°
	R/S	x_2°
	•••	•••

°This value is printed if the PC-100A is used.



SAMPLE PROBLEM

Let's use the program to solve the system of simultaneous equations in our example:

$$2x_1 + 3x_2 + 5x_3 = 12$$
$$4x_1 - 2x_2 + 3.5x_3 = 7.5$$
$$-1.1x_1 + 5.2x_2 - 6.3x_3 = -24$$

Procedure	Press	Display/ Comments	
Fix the decimal point for the desired accuracy (in this problem we'll use 3).	CLR 2nd Fix 3	0.000	
Select the program.	2nd Pgm 02	0.000	
Enter the number of unknowns (here n = 3).	3 A	3.000	
Enter starting column number (j).	1 B	1.000	
Enter each a_{ij} starting with the first column. (Enter each a_{ij} even if a_{ij} is 0.)	2 R/S	2.000	
	4 R/S	4.000	
	1.1 +/- R/S	-1.100	
	3 R/S	3.000	
	2 +/- R/S	-2.000	
	5.2 R/S	5.200	
	5 R/S	5.000	
	3.5 R/S	3.500	
	6.3 +/- R/S	-6.300	
Calculate the determinant.	C	145.850	It will take about 10 - 20 seconds to find the value.
Enter in a 1 to begin the process of entering the b values.	1 D	1.000	
Enter each b_i.	12 R/S	12.000	
	7.5 R/S	7.500	
	24 +/- R/S	-24.000	

Systems of Linear Equations (Master Library Program)

Procedure	Press	Display/Comments
Now we're ready to calculate each x. First begin the routine.	CLR E	1.000 Notice it takes 10-15 seconds to get the display.
Then enter 1 to begin to find each x.	1 2nd A	1.000
Press R/S to find each x.	R/S	−1.525 This is x_1.
	R/S	−0.748 This is x_2.
	R/S	3.459 This is x_3.

Further Explorations

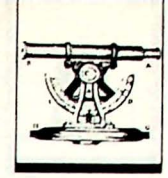
Solve the following systems of simultaneous equations:

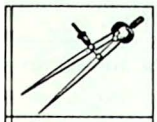
a. $2x_1 - 3.5x_2 + 6x_3 = -1$
$4x_1 + 6x_2 - 3x_3 = 21$
$12x_1 - 3x_2 + 4x_3 = 17$

b. $3x_1 + 4x_3 - 7x_4 = 2.7$
$x_1 + 3x_2 - 7x_3 + 12x_4 = 11.7$
$4x_1 - 2x_2 + 6x_3 + 7x_4 = 21$
$5x_1 + 6x_2 + 9x_4 = 13$

References/ Information *Master Library*, pp. 9-13.

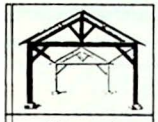
Quadratic Equations





OBJECTIVE

One of the most practical formulas found in algebra is the quadratic formula used to find the roots of a quadratic equation. The following program can be used to find the roots of any quadratic equation, regardless of whether the roots are real or complex.



FACTS/ THEORY

The general form of a quadratic equation is

$$ax^2 + bx + c = 0$$

Where $a \neq 0$. The quadratic formula used to find the roots of the equation is expressed as

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

or

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \text{ and } x_2 = \frac{-b - \sqrt{b_2 - 4ac}}{2a}$$

where x_1 and x_2 are the roots.

The key to whether the roots are real or complex is found in that part of the formula called the *discriminant*,

$$\sqrt{b^2 - 4ac}$$

If $b^2 - 4ac > 0$, both roots are real.
If $b^2 - 4ac = 0$, both roots are real and the same.
If $b^2 - 4ac < 0$, both roots are complex.

When the roots are complex, they can be divided into their real and imaginary parts as follows:

$$x_1 = R + id \text{ and } x_2 = R - id$$

where $R = -b/2a$,
$i = \sqrt{-1}$, and
$d = \dfrac{\sqrt{4ac - b^2}}{2a}$

The following program displays both roots if the roots are real and the values for R and d if the roots are complex.

Quadratic Equations

THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 1 × 2 = STO 4 R/S	Stores a and 2a.
010	2nd Lbl B STO 2 R/S	Stores b.
015	2nd Lbl C STO 3 R/S	Stores c.
020	2nd Lbl D 2nd St flg 1 2nd A'	Sets flag 1 for calculation of x_1, then loops to 2nd A' to calculate $b^2 - 4ac$.
025	2nd Lbl E INV 2nd St flg 1	Unsets flag 1 for calculation of x_2, then loops to 2nd A'
030	2nd Lbl 2nd A' RCL 2 x^2 − 4 × RCL 1 × RCL 3 = 2nd CP INV 2nd x≠t 2nd B' √x 2nd If flg 1 2nd C' +/−	Calculates $b^2 - 4ac$, then tests to see if result < 0. If so, roots are complex and program transfers to 2nd B' to compute R and d. If results ≥ 0, roots are real and program calculates discriminant ($\sqrt{b^2 - 4ac}$). If flag 1 is set, transfers to 2nd C' to compute x_1. If flag 1 not set, changes sign to compute x_2.
053	2nd Lbl 2nd C' − RCL 2 = ÷ RCL 4 = 2nd Prt 2nd Adv R/S	Calculates real root and stops.
066	2nd Lbl 2nd B' √x STO 5 RCL 2 +/− ÷ RCL 4 = 2nd Prt 2nd Adv R/S RCL 5 ÷ RCL 4 = 2nd Prt 2nd Adv R/S	Calculates R and stops. Then computes d and stops.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

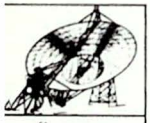
REGISTER CONTENTS

R_{00}		R_{03}	c
R_{01}	a	R_{04}	2a
R_{02}	b	R_{05}	$\sqrt{b^2 - 4ac}$

USING THE PROGRAM

Procedure	Press	Display/Comments
Fix decimal for desired accuracy.	2nd fix n	
Enter data.		
a	A	2a
b	B	b
c	C	c
Calculate x_1.	D	x_1° If root is real. *or* R° Flashing to signify complex roots.
If roots are complex (flashing display), clear error condition and calculate d.	CE R/S	d°
If x_1 is real, calculate x_2.	E	x_2°

°This value is printed if the PC-100A is used.



SAMPLE PROBLEM

1. Find the roots of the equation $2x^2 - 1.5x + 6 = 0$.

Procedure	Press	Display/Comments
Enter program.		0
Fix decimal for desired accuracy.	2nd fix 2	0.00 We'll use 2-place decimal setting for convenience.
Enter a.	2 A	4.00 Display shows 2a.
Enter b.	1.5 +/− B	−1.50
Enter c.	6 C	6.00
Compute x_1.	D	0.38 R—flashing. Indicates roots are complex and R = 0.38.
Calculate value of d.	CE R/S	1.69

The roots are:

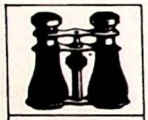
$$x_1 = .38 + 1.69i \text{ and } x_2 = .38 - 1.69i$$

2. Find the roots of the equation $4.1x^2 + 7.5x - 2 = 0$.

Procedure	Press	Display/Comments
Enter a.	4.1 [A]	8.20
Enter b.	7.5 [B]	7.50
Enter c.	2 [+/−] [C]	−2.00
Compute x_1.	[D]	0.24 Real root.
Compute x_2.	[E]	−2.07

The roots are:

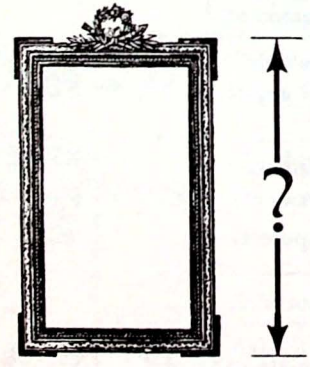
$$x_1 = 0.24 \text{ and } x_2 = -2.07.$$



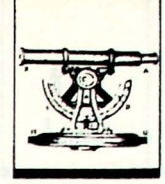
FURTHER EXPLORATIONS

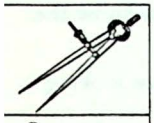
1. Find the roots of the equation $3x^2 = 15$. (*Answer:* $x_1 = 2.24$, $x_2 = -2.24$.)
2. The frame around a square picture is 4 cm (approximately 1.5748 inches) wide. If the area of the picture is one-half the total area (picture and frame), what are the dimensions of the frame? (*Answer:* 27.31 cm by 27.31 cm or 10.75 in. by 10.75 in.)

REFERENCES/ INFORMATION See the quadratic equation program in *Personal Programming*, pp. IV-79—IV-83.



Trig and Triangles



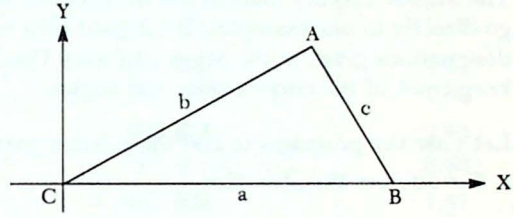


OBJECTIVE

The study of triangles has fascinated mankind since early Greek mathematicians first began explaining them. Many useful and practical discoveries have come about through the analysis of this interesting shape. The following programs from the Master Library incorporate the Law of Sines, the Law of Cosines, Heron's Formula, and the relationships of angles to help us explore triangles and find missing parts of triangles.

FACTS/ THEORY

The sum of the three angles of a triangle equals 180°. This simple fact is very useful when searching for lengths of sides and measures of angles of a triangle. If we start with the triangle



and designate each angle by its vertex, the following laws apply:

The Law of Cosines: $c^2 = a^2 + b^2 - 2ab \cos C$

or $a^2 = b^2 + c^2 - 2bc \cos A$

or $b^2 = c^2 + a^2 - 2ac \cos B$

The Law of Sines: $\frac{\sin A}{a} = \frac{\sin B}{b} = \frac{\sin C}{c}$

Heron's Formula proves useful for finding areas of triangles when the lengths of the three sides are known.

$$\text{Area} = \sqrt{s(s-a)(s-b)(s-c)}$$

$$\text{where } s = \frac{a+b+c}{2}$$

Program ML-11 will be used to find the remaining angles and sides if you know the following:

SSS (You know all three sides and are looking for the measurements of the angles.)

SSA (You know two sides and a non-inclusive angle and are looking for the other side and angles.)

SAS (You know two sides and the inclusive angle and are looking for the other side and the other angles.)

Program ML-12 will be used to find missing angles and sides if you know the following:

ASA (You know two angles and the inclusive side and are looking for the other two sides and the third angle.)

SAA (You know two angles and a non-inclusive side and are looking for the other sides and angle.)

Programs ML-11 and ML-12 will also help you find the area of the triangle if the lengths of the three sides are known.

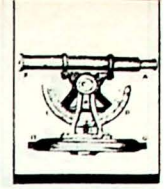
The Master Library manual has instructions for using the programs, so we'll go directly to our examples. It's a good idea to follow the side and angle designations given in the Master Library User Instructions to make sure you keep track of the correct sides and angles.

SAMPLE PROBLEM

Let's use the programs to find the missing parts of the following triangles:

1. C = 30°, a = 45m, b = 42m

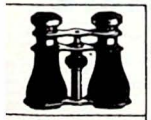
Procedure	Press	Display/Comments
Select the program desired.	CLR 2nd Pgm 11	0.
Fix the decimal to desired accuracy (we'll use two places).	2nd Fix 2	0.00
Instruct the program to begin.	2nd E'	0.00
Enter a.	45 A	45.00
Enter b.	42 B	42.00
Enter C.	30 C	30.00
Calculate c.	E	22.70 Side c is 22.7 m.
Calculate B.	2nd B'	67.67 B is 67.67°.
Calculate A.	2nd C'	82.33 A is 82.33°.



2. $B=\frac{\pi}{3}$, $C=\frac{\pi}{6}$, $a=11.1$ cm

Find the area, also.

Procedure	Press	Display/Comments	
Select the program.	CLR 2nd Pgm 12	0.00	
Select radian mode.	2nd Rad	0.00	
Enter a.	11.1 A	11.10	
Enter B.	2nd π	3.14	
	÷ 3 = B	1.05	
Enter C.	2nd π	3.14	
	÷ 6 = C	0.52	
Calculate A.	2nd A'	1.57	A is 1.57 radians.
Calculate b.	D	9.61	Side b is 9.61 cm.
Calculate c.	E	5.55	Side c is 5.55 cm.
Find the area. (Since we've calculated the missing parts, the lengths of sides a, b, and c are already stored in the proper memories.)	2nd C'	26.68	The area is 26.68 cm^2.



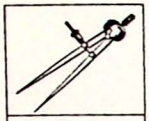
FURTHER EXPLORATIONS

1. One side of a regular pentagon inscribed in a circle is 10 cm long. What is the radius of the circle? (*Answer:* 8.51 cm)
2. The angle of elevation of a radio tower on the top of a building is 45° (at the base of the tower). The angle of elevation of the tower is 60° (at the top of the tower). Both angles were taken from a point on the ground 75 m away from the base of the building. How tall is the building? How tall is the tower? (*Answers:* 75 m, 54.90 m)
3. A navigator locates his position, A, on a map. He is 75 km due west of city B and 94 km from city C. The distance between B and C is 38 km. Find angle A (in degrees). (*Answer:* 22.60°)

REFERENCES/ INFORMATION

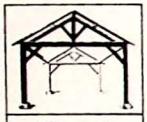
Master Library, pp. 34–44.

2 Trig Functions



OBJECTIVE

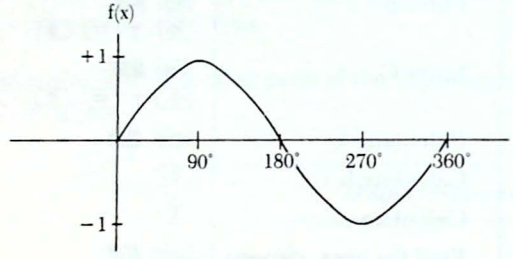
Trigonometric functions are periodic and their graphs resemble waves. We are daily being bombarded by waves of all kinds, sound waves, ultraviolet rays, infrared waves, and many types of electromagnetic waves. Graphing and understanding trig functions can provide not only practical information but also much aesthetic pleasure. The following program can help you graph the sine, cosine, and tangent functions, as well as help you explore periodicity, phase shifts, amplitudes, and translations of trig functions.



FACTS/ THEORY

The graphs of trig functions follow repetitive patterns. Understanding the various aspects of these patterns can help simplify graphing the functions.

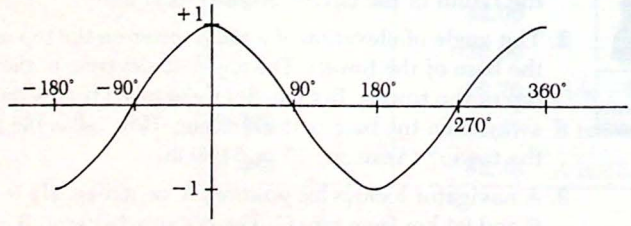
To exemplify some of the various aspects of trig functions, let's look at a simple sine curve.



$f(x) = \sin x$

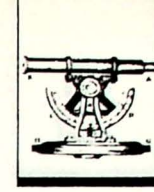
Notice that in this case the maximum f(x) value is $+1$ and the minimum f(x) value is -1. One half the vertical distance between the maximum and minimum points is called the *amplitude* of the wave. Notice that every 360° the wave repeats itself. The value P where $f(x+P) = f(x)$ (in this case 360°) is called the *period* of the function.

Another graph that will show us some useful properties is $f(x) = \sin(x+90°)$.

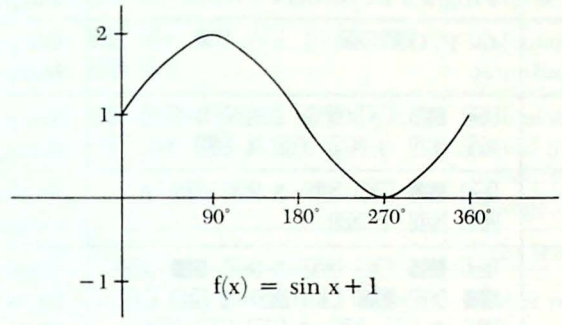


$f(x) = \sin(x+90°)$

The graph of $f(x) = \sin(x+90°)$ looks like the graph of $f(x) = \sin x$ *except* for a shift to the left. This shift is called a *phase shift.*



One more graph which is useful in showing shift patterns is $f(x) = \sin x + 1$.



The graph of $f(x) = \sin x + 1$ looks like the graph of $f(x) = \sin x$ except for an upward shift of one unit.

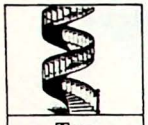
Generally speaking, the following family of functions can help you better understand what's going on:

$$f(x) = a \sin [b (x - c)] + d$$

where the *amplitude* of the function is $|a|$, the *period* of the function is $\frac{360^\circ}{b}$, a *phase shift* left or right takes place depending upon the value of c (*left* if $c<0$, *right* if $c>0$), and a *shift up* or *down* depends on d (*up* $|d|$ units if $d>0$, *down* $|d|$ units if $d<0$).

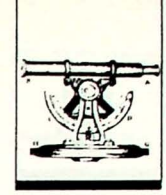
The ideas on amplitudes, periods, and phase shifts can be similarly applied to the other trig functions. *Note:* Basically, circular functions are very closely related to trig functions. Whereas the domains of circular functions rely on the unit circle and π, the domains of trigonometric functions are angular. For simplicity's sake, we are working with trig functions and use angular measurement. However, the program which follows can be changed to radian measure by simply pressing 2nd Rad before you use the program and inserting radian measures for angular measures.

The following program allows you to enter the values of a, b, c, and d, and m, n, and i (where m is an initial x value, n is a final x value, and i is the value of an increment) and then to calculate and display the points (m, f(m)), (m + i, f(m + i)), . . . (n, f(n)) for $f(x) = a [\sin b(x - c)] + d$, $f(x) = a \cos [b(x - c)] + d$, or $f(x) = a \tan [b(x - c)] + d$. Once the values are calculated, you can graph the functions and explore the results of changes in periodicity, amplitude, or phase shifts.



THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 1 R/S STO 2 R/S STO 3 R/S STO 4 R/S	Stores a, b, c, and d in memories 1–4 respectively.
014	2nd Lbl B STO 5 R/S STO 6 R/S STO 7 R/S	Stores m, n, and i in memories 5, 6, and 7.
025	2nd Lbl C RCL 5 2nd Pause 2nd Pause 2nd Prt (RCL 2 × (RCL 5 − RCL 3)) 2nd sin × RCL 1 + RCL 4 = 2nd Prt 2nd Adv R/S SBR x^2	Starts program to evaluate $f(x) = a \sin [b(x-c)] + d$. Ordered pairs are displayed (and printed if the PC-100A is used; otherwise, print commands are ignored). Subroutine x^2 increments x by i.
057	INV 2nd $x \geq t$ $\sqrt{x}$ GTO C	Tests new x value generated by subroutine x^2. If $n - x < 0$, program transfers to subroutine $\sqrt{x}$; otherwise it loops back to C.
062	2nd Lbl x^2 RCL 7 SUM 5 RCL 6 − RCL 5 = INV SBR	Increments x by i and subtracts new x from n.
075	2nd Lbl $\sqrt{x}$ CLR R/S	If $n - x < 0$, display is cleared and program run stops.
079	2nd Lbl D RCL 5 2nd Pause 2nd Pause 2nd Prt (RCL 2 × (RCL 5 − RCL 3)) 2nd cos × RCL 1 + RCL 4 = 2nd Prt 2nd Adv R/S SBR x^2 INV 2nd $x \geq t$ $\sqrt{x}$ GTO D	Calculates and displays ordered pairs (x, f(x)) from $x = m$ to $x = n$ step i for $f(x) = a \cos[b(x-c)] + d$.
116	2nd Lbl E RCL 5 2nd Pause 2nd Pause 2nd Prt (RCL 2 × (RCL 5 − RCL 3)) 2nd tan × RCL 1 + RCL 4 = 2nd Prt 2nd Adv R/S SBR x^2 INV 2nd $x \geq t$ $\sqrt{x}$ GTO E	Calculates and displays ordered pairs from $x = m$ to $x = n$ step i for $f(x) = a \tan [b(x-c)] + d$.



Step #	Program Keystrokes	Purpose/Comments
153	2nd Lbl 2nd A' RCL 1 2nd |x| 2nd Prt R/S	Calculates and displays the *amplitude.*
160	2nd Lbl 2nd B' RCL 2 1/x × 360 = 2nd Prt R/S	Calculates and displays the *period* in degrees. If radian measurement is desired, change 360 to 2 π.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

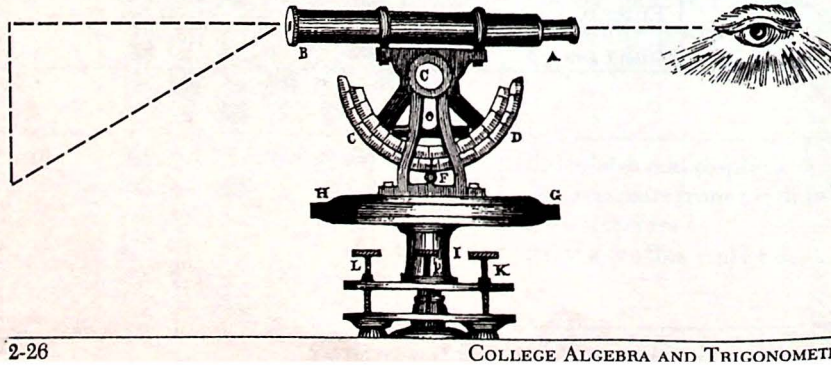
R_{00}	R_{03} c	R_{06} n
R_{01} a	R_{04} d	R_{07} i
R_{02} b	R_{05} m	

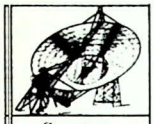
USING THE PROGRAM

Procedure	Press	Display/Comments
Fix decimal point to desired accuracy (n places).	2nd Fix n	In most situations, n = 2 for our desired accuracy.
Choose the desired angular mode.	2nd Deg. 2nd Rad 2nd Grad	If degrees are used. If radians are used. If grads are used.
Enter a.	A	a
Enter b.	R/S	b
Enter c.	R/S	c
Enter d.	R/S	d
Enter m.	B	m
Enter n.	R/S	n
Enter i.	R/S	i

Procedure	Press	Display/Comments
Choose the desired program run.		
1. f(x) = a sin[b(x − c)] + d	C	m° f(m)°
Press R/S after each f(x).	R/S	m + i° f(m + i)°
	. . .	. . .
	R/S	n° f(n)°
2. f(x) = a cos [b(x − c)] + d	D	m° f(m)°
	R/S	m + i° f(m + i)°
	. . .	. . .
	R/S	n° f(n)°
3. f(x) = a tan [b(x − c)] + d	E	m° f(m)°
	R/S	m + i° f(m + i)°
	. . .	. . .
	R/S	n° f(n)°
4. Amplitude (|a|).	2nd A′	|a|°
5. Period (P).	2nd B′	P°

°This value is printed if the PC-100A is used.

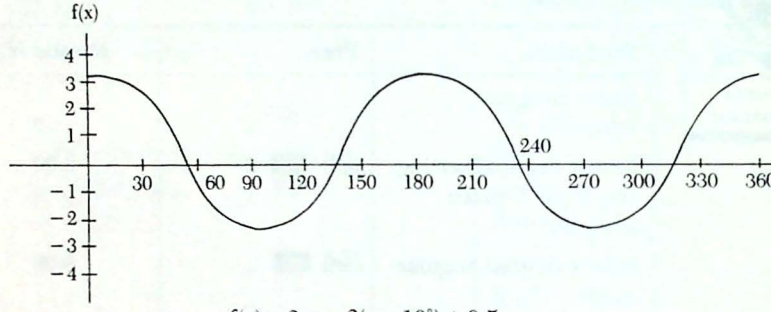




SAMPLE PROBLEM

a. Graph the trig function $f(x) = 3 \cos 2[(x - 10^\circ)] + 0.5$ for $x = 0^\circ$ to $x = 360^\circ$ with $i = 30^\circ$.

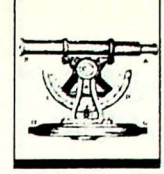
Procedure	Press	Display/Comments	
Enter program carefully.		0	
Select decimal setting (we'll use 2-place accuracy).	2nd Fix 2	0.00	
Select desired angular mode.	2nd Deg	0.00	
Enter a, b, c, and d.			
$a = 3$	3 A	3.00	
$b = 2$	2 R/S	2.00	
$c = 10$	10 R/S	10.00	
$d = 0.5$	.5 R/S	0.50	
Enter m, n, and i.			
$m = 0$	0 B	0.00	
$n = 360$	360 R/S	360.00	
$i = 30$	30 R/S	30.00	
Select desired program run: $f(x) = a \cos [b(x - c)] + d$	D	0.00 3.32	m, for 2 pauses. f(m)
Press R/S after each f(x).	R/S	30.00	m + i, for 2 pauses.
		2.80	f(m + i)
	...	...	
	R/S	360.00	n, for 2 pauses.
		3.32	f(n)
	R/S	0	Indicates value of n has been reached.



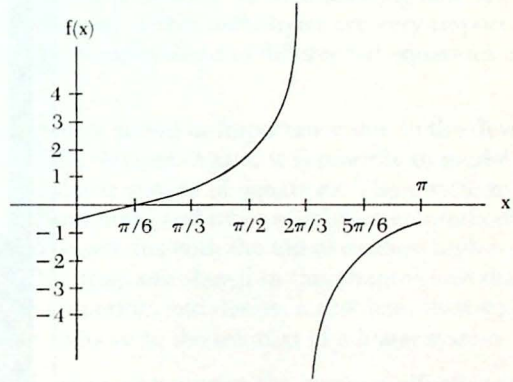
$f(x) = 3 \cos 2(x - 10^\circ) + 0.5$

b. Graph the trig function $f(x) = \tan\left(x - \frac{\pi}{6}\right)$ for $x = 0$ to $x = \pi$ with $i = \frac{\pi}{6}$.

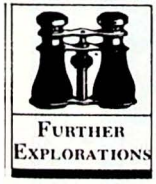
Procedure	Press	Display/Comments
Select the desired angular mode.	CLR 2nd Rad	0.00
Enter a, b, c, and d.		
$a = 1$	1 A	1.00
$b = 1$	1 R/S	1.00
	2nd π ÷	3.14
$c = \pi/6$	6 = R/S	0.52
$d = 0$	0 R/S	0.00
Enter m, n, and i.		
$m = 0$	0 B	0.00
$n = \pi$	2nd π R/S	3.14
$i = \pi/6$	÷ 6 = R/S	0.52
Select the desired program run: $f(x) = a \tan[b(x - c)] + d$.	E	0.00 m −0.58 f(m)
	R/S	0.52 m + i 0.00 f(m + i)
	R/S	1.05 0.58



Procedure	Press	Display/Comments
	R/S	1.57
		1.73
	R/S	2.09
		1.13 12
	R/S	2.62
		−1.73
	R/S	3.14 n
	R/S	−0.58 f(n)
	R/S	0 x=n



$f(x) = \tan (x - \pi/6)$



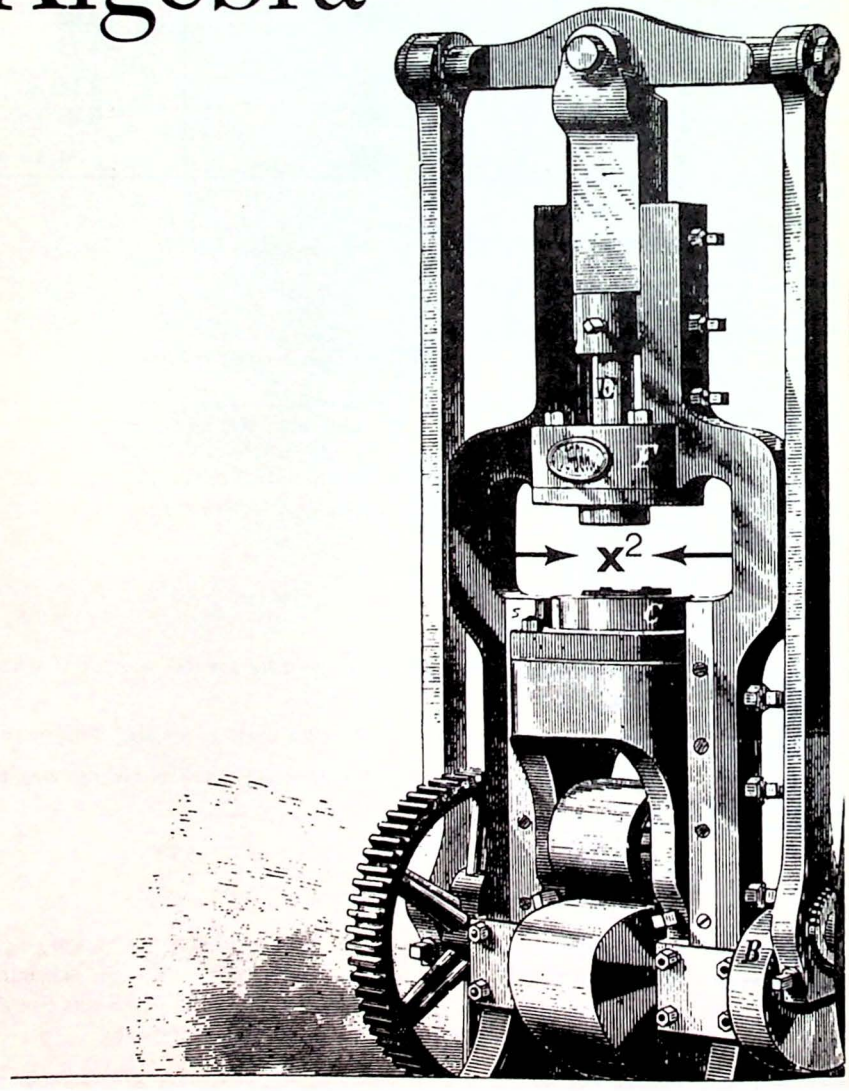
FURTHER EXPLORATIONS

1. Graph the function $f(x) = 3 \sin x - 4$ for $x = 180^\circ$ to $x = 360^\circ$ with increments of 20°.
2. Graph the function $f(x) = \tan 2x$ for $x = 0^\circ$ to $x = 180^\circ$ with increments of 10°.
3. Graph the sine function whose period is $2\pi/3$ and whose amplitude is 4.
4. Show, by graphing, that the $\sin (-x) = -\sin x$.
5. Show, by graphing, that the $\cos (x \pm \pi) = -\cos x$.
6. Graph $2 \sin (\pi/2 - x)$ and $2 \cos x$ for $x = 0$ to $x = 2\pi$.

REFERENCES/ INFORMATION

Most elementary trigonometry texts cover this topic. Also, if you are using the PC-100A printer, you may be interested in writing in subroutines commanding the printer to plot the generated data for you. See *Personal Programming*, p. VI-10, for a discussion of plotting data.

3 Calculus and Linear Algebra



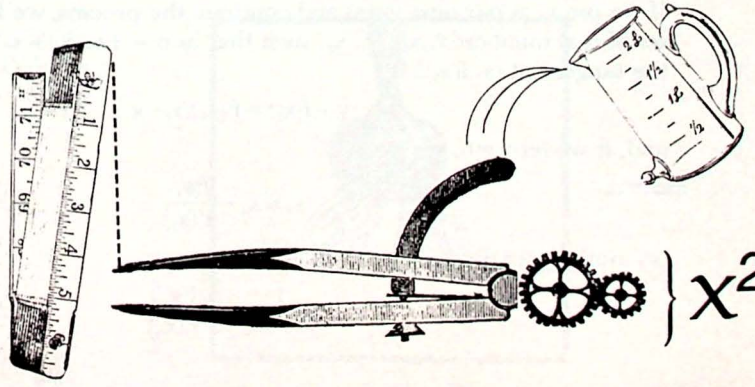
Introduction

Calculus and linear algebra form an important part of the mathematical foundation from which scientists and engineers develop their special areas. Using these mathematical tools to solve particular problems can provide a deeper understanding of the basic ideas or concepts involved in these areas. The material presented in this chapter represents only a small segment of the numerical methods associated with calculus and linear algebra and their applications to the programmable calculator, but these can give you a basis from which to explore the whole area of numerical analysis.

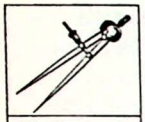
Many extremely complex problems in today's scientific research are approached through mathematical models involving large sets of differential equations. The concepts of differentiation and integration from the calculus are examined in several of the programs here, not only illustrating the computational and limiting processes, but also showing how solution techniques can be developed. These techniques are very important, since new methods for solving these systems of differential equations are constantly evolving.

Linear algebra has recently grown in importance due to the development of high-speed computational devices. Again, it is possible to model many problems in terms of a linear system of equations. These systems can be overwhelming in size, and more and more sophisticated methods are being used today to solve these systems with the aid of modern high-speed computing devices. The programs shown in this chapter introduce one of the basic techniques, row reduction, and discuss a problem, least-squares approximation, which reduces to the solution of a linear system.

These examples, while they only suggest the many applications that are possible, should give you a place to begin your own exploration.

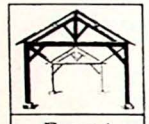


3 Approximating Roots of a Function



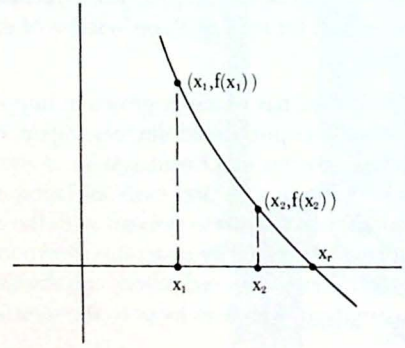
OBJECTIVE

One way to approximate real roots to an equation of the form $f(x)=0$ involves a technique called the Newton-Raphson method. This method makes efficient use of the iterative capabilities of the programmable calculator.



FACTS/ THEORY

If we are given that $f(x)$ is differentiable on $[x_a,x_b]$, that x_a is "close" to x_b, and that $f(x_a)$ and $f(x_b)$ have different signs, $f(x)=0$ will usually have one root, x_r, between x_a and x_b. Basically the Newton-Raphson method assumes some approximation to the root (we'll call it x_1), then replaces the curve $y=f(x)$ by its tangent at $(x_1,f(x_1))$, and finds the point $(x_2,0)$ where the tangent intersects the x-axis. Generally, x_2 will be closer to x_r than x_1.



If we use x_2 as our next guess and continue the process, we'll usually derive a set of real numbers $x_1,x_2,\ldots x_n$, such that as $n \to +\infty$, $x_n \to x_r$. The equation of the tangent at $(x_1,f(x_1))$ is

$$y - f(x_1) = f'(x_1)(x_2 - x_1)$$

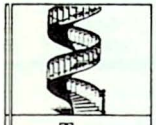
and, if we let $y=0$, we get

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

or, more generally,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

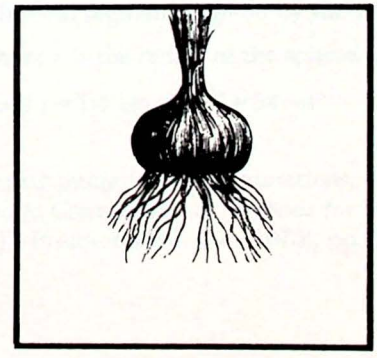
The following program allows you to enter special subroutines for f(x) and f′(x) and then approximates x_r for any given n, using the Newton-Raphson method.



The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 01 0 x⇄t CLR R/S	Enters x_1 and clears the t register.
008	2nd Lbl B STO 2 R/S	Enters n.
013	2nd Lbl D RCL 1 − E ÷ 2nd E′ = STO 1 2nd Dsz 2 D RCL 1 2nd Prt R/S	Calculates $x_n - \frac{f(x_n)}{f'(x_n)}$, and stores the new "guess" in memory 1. Then decrements memory 2 by 1 and checks to see if memory 2 = 0. If not, loops back to calculate new approximation.
031	2nd Lbl E	This is where you'll add your subroutines for f(x) and f′(x).
	LRN RST	Exits learn mode.

Note: Print command may be omitted if the PC-100A is not used.



3

Approximating Roots of a Function

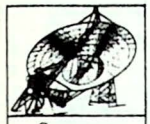
Using the Program

Procedure	Press	Display/Comments
Transfer the program to subroutine [E] and enter program steps for f(x) and f′(x). Use [RCL] 01 wherever you need the x value.	[GTO] [E] [LRN] Program steps for f(x) [INV] [SBR] [2nd] [Lbl] [2nd] [E′]	033 00 "Goes to" subroutine [E].
Note: the last program step in each subroutine *must* be [INV] [SBR].	Program steps for f′(x) [INV] [SBR] [LRN]	0. Exits learn.
Enter first approximation.	x_1 [A]	0.
Enter n.	n [B]	n.
Execute program.	[D]	x_n° This value should be close to x_r.

°This value is printed if the PC-100A is used.

Register Contents

R_{00}
R_{01} x_n
R_{02} Used



Sample Problem

Using the Newton-Raphson method, find a root for $f(x) = x^2 - 5$. First let $n = 3$ and then $n = 5$ to show how quickly x_n approaches x_r. (Notice that $0 = x^2 - 5$ has a root at $x = \sqrt{5}$.) The derivative of $x^2 - 5$ is $2x$; therefore, $f'(x) = 2x$.

Procedure	Press	Display/Comments
Enter program.		0
Transfer to subroutine [E] and enter subroutines for f(x) and f′(x).	[GTO] [E] [LRN] [(] [RCL] 1 [x^2] [−] 5 [)] [INV] [SBR] [2nd] [Lbl] [2nd] [E′] [(] [RCL] 1 [×] 2 [)] [INV] [SBR] [LRN]	033 00 050 00 0.
Enter x_1 (we'll use 1 as our first "guess").	1 [A]	0
Enter n.	3 [B]	3.
Calculate x_r.	[D]	2.238095238 x_r,n = 3
Now repeat process for n = 5.	1 [A] 5 [B] [D]	0 5. 2.236067978 x_r,n = 5

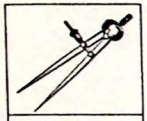
Further Explorations

1. Find the roots of $2 \cos x = x^2$ (correct to four decimal places).
2. Find all the roots of $x^3 + 2x^2 - 1 = 0$ (correct to three decimal places).
   Hint: Be careful when using the [y^x] key with negative values.
   (*Answers:* −1.618, −1, 0.618)
3. The volume of a spherical segment is given by the formula $V = \frac{(\pi h^2)}{3}(3r - h)$, where r is the radius of the sphere and h is the height of the segment. Find h if $r = 1.5$ cm and $V = 3\pi \text{cm}^3$.

References/ Information

See any textbook covering numerical approximations, such as Samuel S. McNeary, *Introduction to Computational Methods for Students of Calculus* (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1973), pp. 110-115.

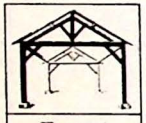
3 Numerical Approximation of Derivatives



OBJECTIVE

The concept of the derivative is of fundamental importance in mathematics and in its application to real world problems. The derivative, in fact, measures the rate of change of most things that can be described mathematically. For instance, the acceleration of an object can be determined by taking the derivative of its velocity, etc.

Numerical approximation of a derivative is often less complex than the actual mechanics of the derivation process itself and can be sufficiently accurate in most cases. Sometimes the only way to find the derivative of a function is through numerical approximation. Numerical methods are also useful for simple functions when a large number of values are to be evaluated.



FACTS/ THEORY

Numerical methods of approximating derivatives evaluate the mathematical function, f(x), at discrete intervals, h. The smaller the sampling interval, the more accurate is the approximation of the derivative. The following relationship is used for the approximation:

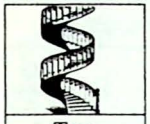
$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

where $f'(x)$ is the derivative of $f(x)$.

The following program can calculate the sample interval, h, for you, or you can choose it yourself. The calculated value of h is based on the magnitude of x as shown below.

$$h = \begin{cases} (.001)\,|x| & \text{if } |x| > 0.01 \\ .00001 & \text{if } |x| \leq 0.01 \end{cases}$$

These values for h have been experimentally determined to work reasonably well. Its applicability is certainly not universal, however, and it should be used cautiously. You may want to try different h values to be sure you get the degree of accuracy required by the problem.



THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [A] [STO] 1 [STO] 2 [R/S]	Stores values of x.
007	[2nd] [|x|] [x:t] .01 [2nd] [x≥t] [C] .001 [×] [RCL] 1 [2nd] [|x|] [=] [B]	Calculates h, based on magnitude of x. If $\|x\| \leq .01$, transfers to [C]. If $\|x\| > .01$, calculates h as .001 |x|.
024	[2nd] [Lbl] [C] .00001	Enters .00001 as h.
032	[2nd] [Lbl] [B] [STO] 03	Stores h.
036	[SUM] 1 [E] [STO] 4 [RCL] 2 [−] [RCL] 3 [=] [STO] 1 [E] [+/−] [+] [RCL] 4 [=] [÷] [(] 2 [×] [RCL] 3 [=] [2nd] [Prt] [R/S]	Calculates f(x+h), f(x−h), and $\frac{f(x+h)-f(x-h)}{2h}$ to find f′(x).
064	[2nd] [Lbl] [E]	Labels subroutine for calculation of f(x).
	[LRN] [RST]	Exits learn mode.

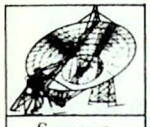
Note: Print command may be omitted if the PC-100A is not used.

REGISTER CONTENTS			
R_{00}		R_{03}	h
R_{01}	(x+h),(x−h)	R_{04}	f(x+h)
R_{02}	x	R_{05}	

USING THE PROGRAM

Procedure	Press	Display/Comments
Set decimal for desired accuracy.	2nd fix n	
Enter keystrokes for f(x). Begin and end the calculation with parentheses, rather than using the = key, and recall memory register 1 wherever the value of x is needed.	GTO E LRN Program steps for f(x).	066 00 Enters learn mode at subroutine E .
When all steps are entered, end the subroutine with INV SBR to insure proper return to main program.	INV SBR	
Exit learn mode and reset program.	LRN RST	0.
Enter x.	A	x
Enter h,	B	
OR	*OR*	f'(x)°
Calculate h.	R/S	

°This value is printed if the PC-100A is used.

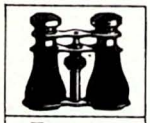


SAMPLE PROBLEM

Calculate $f'(.1)$ and $f'(2.1)$ for $f(x) = 3x^3 - 2x + 1$. Let the calculator select h.

Procedure	Press	Display/Comments
Enter program.		0
Select two-place decimal.	2nd fix 2	0.00
Enter program for f(x).	GTO E LRN (3 × RCL 1 y^x 3 − 2 × RCL 1 + 1) INV SBR	066 00
Exit learn mode and reset.	LRN RST	0.00
Enter first x value.	.1 A	0.10
Calculate $f'(.1)$.	R/S	−1.91
Enter second x value.	2.1 A	2.10
Calculate $f'(2.1)$.	R/S	37.69

In both of these cases $(.001)|x|$ was used for h. You get the same results by manually entering this value for h. To key in a new f(x), press GTO E LRN and enter the new function keystrokes. Don't worry about writing over all the old f(x) locations as long as the new f(x) is ended with) INV SBR .



FURTHER EXPLORATIONS

1. If $f(x) = \dfrac{4x^3}{1+x^3}$, find $f'(.3)$ and $f'(0)$.

   (*Answers:* $f'(.3) = 1.023959895$; $f'(0) = .0000000004$)

2. If $f(x) = \sin x$, find sufficient values of $f'(x)$ over the interval $(0,\pi)$ to plot the graph of $f'(x)$.

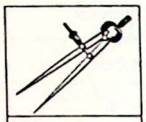
3. Devise a program to compute $f''(x)$ in addition to $f'(x)$ by using the following approximation:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

REFERENCES/ INFORMATION

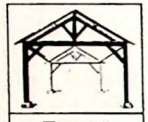
H. P. Greenspan and D. J. Benney, *Calculus—An Introduction to Applied Mathematics* (New York: McGraw-Hill, 1973), pp. 177-183.

3 Definite Integrals– Trapezoidal Rule



OBJECTIVE

One of the most useful aspects of a programmable calculator is the ability of the machine to perform iterative processes. Approximating definite integrals is an area of calculus where iterative processes are especially useful. One method, Simpson's Rule, can be found in the *Master Library* (ML-09,10). The following program uses another common iterative method called the "Trapezoidal Rule" for approximating definite integrals.



FACTS/ THEORY

The trapezoidal rule works under the supposition that f(x) is continuous for the area of interest, $x_0 \leq x \leq x_n$. If we divide $[x_0, x_n]$ into n equal intervals of length $h = \frac{(x_n - x_0)}{n}$, then

$$\int_{x_0}^{x_n} f(x)dx \approx \frac{h}{2}\left[f(x_0) + 2f(x_1) + \ldots + 2f(x_{n-1}) + f(x_n) \right]$$

where $x_1 = x_0 + h$

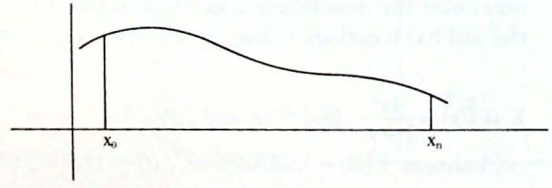
$x_2 = x_0 + 2h$

$x_3 = x_0 + 3h$

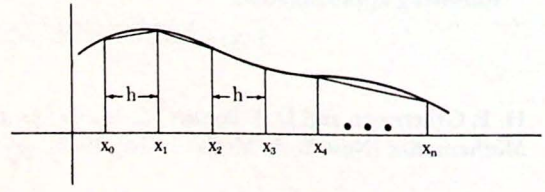
$\cdots$

$x_n = x_0 + nh$

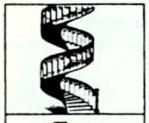
Assume that f(x) has the following graphic representation from x_0 to x_n:



The trapezoidal rule breaks the area under the curve into n trapezoids (each having a "height" of h) as shown below.



The more intervals under the curve, the more accurate is the approximation of the area of interest. Using several different values of n for approximating an f(x) can show the degree of accuracy you have.



THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 1 R/S	Stores x_0.
005	2nd Lbl B STO 2 R/S	Stores x_n.
010	2nd Lbl C STO 3 R/S	Stores n.
015	2nd Lbl D CLR STO 6 STO 7 (RCL 2 − RCL 1) ÷ RCL 3 = STO 04	Calculates h and stores in register 4.
035	RCL 1 E SUM 07	Calculates $f(x_0)$ and adds to register 7.
040	RCL 3 STO 5 2nd Lbl 2nd A' 2nd Dsz 5 2nd B' RCL 2 E SUM 7 RCL 7 × RCL 4 ÷ 2 = 2nd Prt R/S	Calculates total integral.
064	2nd Lbl 2nd B' RCL 4 SUM 6 RCL 6 E × 2 = SUM 7 GTO 2nd A'	
080	2nd Lbl E	Labels f(x) subroutine.
	LRN RST	Exits learn mode.

Note: Print command may be omitted if the PC-100A is not used.

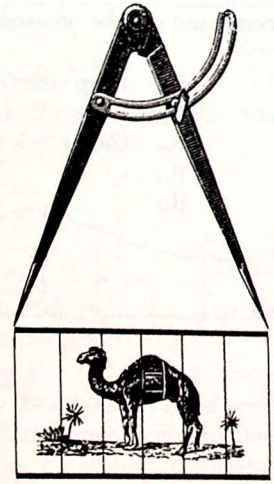
REGISTER CONTENTS

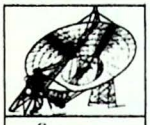
R_{00}		R_{05}	Loop counter
R_{01}	x_0	R_{06}	Used
R_{02}	x_n	R_{07}	Used
R_{03}	n	R_{08}	x_i
R_{04}	h	R_{09}	

3 Definite Integrals—Trapezoidal Rule

Using the Program

Procedure	Press	Display/Comments
Position program pointer for f(x) subroutine entry.	GTO E LRN (Program keystrokes for f(x))	082 00
Enter keystrokes for f(x), assuming that x is in the display. If necessary, store x in register 08. Begin subroutine with a (and end with) INV SBR. Do not use =.	INV SBR LRN RST	0. Exits learn mode.
Enter x_0.	A	x_0
Enter x_n.	B	x_n
Enter n.	C	n
Approximate integral.	D	$\int_{x_0}^{x_n} f(x)\,dx$°
To evaluate for different n, repeat last two steps.		



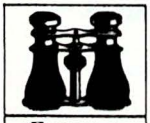


SAMPLE PROBLEM

Approximate the integral $\int_0^5 (x^2 + 2x - 1)\,dx$, first using 5 and then 10 intervals.

Procedure	Press	Display/Comments
Enter program.		**0**
Go to label [E] and enter f(x) subroutine.	[GTO] [E] [LRN] [(] [STO] 8 [x^2] [+] 2 [×] [RCL] 8 [−] 1 [)] [INV] [SBR] [LRN]	**082 00** **0.**
Enter x_0.	0 [A]	**0.**
Enter x_n.	5 [B]	**5.**
Enter n.	5 [C]	**5.**
Approximate integral using 5 intervals.	[D]	**62.5** f(x) for 5 intervals. (Allow about 15 sec for calculation.)
Approximate integral using 10 intervals.	10 [C] [D]	**61.875** f(x) for 10 intervals.

The approximation for 100 intervals is 61.66875, so the 10-interval approximation is fairly accurate. The value of n can be selected to yield the accuracy you need.



FURTHER EXPLORATIONS

1. Approximate $\int_0^2 e^{x^2} dx$. (Answer will vary, depending on the value of n.

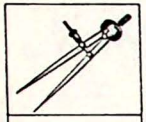
   For example, if n = 20, the approximation will be 16.63395884.)

2. Approximate $\int_0^{2\pi} \sin x\,dx$, using n = 10. Remember to use the radian mode here. (*Answer:* 0.)

REFERENCES/ INFORMATION

Most calculus texts discuss definite integrals in some detail. For example, see Lynn Loomis, *Calculus* (Reading, Mass.: Addison-Wesley Publishing Company, 1974), chapter 10.

3 Runge-Kutta Method (Systems of Differential Equations)



OBJECTIVE

Many methods of numerical solution for differential equations exist. The Euler Method discussed in the Physics chapter is reasonably simple to apply to ordinary differential equations, but the error produced at each approximation step is large unless the step size is very small. Therefore, we seek a method which will produce more accuracy but at the expense of increased computation time. Two such methods are Runge-Kutta and Predictor-Corrector. The Predictor-Corrector Method is discussed in conjunction with the Sedimentation problem in the Biology chapter.

In this program the Runge-Kutta Method is applied to a system of ordinary differential equations. The program included here will solve a system of one or two differential equations. The ideas used in developing these methods are fundamental to an entire class of integration methods for differential equations, and will help to provide both technique and theoretical understanding.

FACTS/ THEORY

The technique behind any numerical solution method is to find a set of points (x_j, y_j) through which the solution passes, or almost passes, since only an approximation is found. The x_j values are usually specified by setting a step size, Δx, and letting

$$x_{j+1} = x_j + \Delta x \qquad j = 0, 1, 2, \ldots$$

The y_j values are determined by the numerical scheme with the initial value y_0 used as a starting point.

The Runge-Kutta Method computes y_{j+1} by using the value of y_j and the first few terms in the Taylor series expansion for $y(x)$ about x_j. The difference between the value of y at x_{j+1} and x_j can be expressed as

$$y_{j+1} - y_j = a\Delta x\, f(x_j, y_j) + b\Delta x\, f(x_j + \alpha\Delta x,\ y_j + \beta\Delta y) \qquad [1]$$

where $\Delta y = \Delta x f(x_j, y_j)$ and a, b, α, and β are constants to be determined.

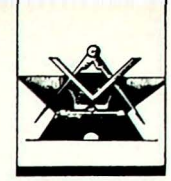
The Taylor series for y_{j+1} is

$$y_{j+1} = y_j + \Delta x\, y'_j + \frac{(\Delta x)^2}{2} y''_j + \ldots$$

The double Taylor series for $f(x_j + \alpha\Delta x, y_j + \beta\Delta y)$ about (x_j, y_j) is

$$f(x_j + \alpha\Delta x, y_j + \beta\Delta y) = f(x_j, y_j) + \alpha\Delta x \frac{\partial f}{\partial x}(x_j, y_j) + \beta\Delta y \frac{\partial f}{\partial y}(x_j, y_j)$$

$$+ 1/2 \left\{ \alpha^2(\Delta x)^2 \frac{\partial^2 f}{\partial x^2}(x_j, y_j) + 2\alpha\beta\Delta x\Delta y \frac{\partial^2 f}{\partial x \partial y}(x_j, y_j) + \beta^2(\Delta y)^2 \frac{\partial^2 f}{\partial y^2}(x_j, y_j) \right\} + \ldots$$



This can be substituted into the right and left side of equation [1]. Setting the coefficients of 1, Δx, and $(\Delta x)^2$ on each side of the equation equal yields the following relations for a, b, α, and β.

$$a+b=1$$
$$b\alpha = b\beta = 1/2$$

One choice of these constants is

$$a=0 \qquad b=1$$
$$\alpha = 1/2 \qquad \beta = 1/2$$

Choosing these constant values, we have the following formula for y_{j+1} in terms of y_j and x_j.

$$y_{j+1} = y_j + \Delta x f(x_j + \frac{\Delta x}{2}, y_j + \frac{\Delta x f(x_j, y_j)}{2})$$

This is called a second-order Runge-Kutta formula, also called an open Newton-Cotes formula.

Higher order methods can also be developed using the same procedure, simply by satisfying equation [1] to higher terms in Δx. For example, equating terms up to $(\Delta x)^4$ yields a fourth-order method with the approximating formula

$$y_{j+1} = y_j + 1/6\,(K_1 + 2K_2 + 2K_3 + K_4)$$

where
$$K_1 = \Delta x f(x_j, y_j)$$
$$K_2 = \Delta x f(x_j + \frac{\Delta x}{2}, y_j + \frac{K_1}{2})$$
$$K_3 = \Delta x f(x_j + \frac{\Delta x}{2}, y_j + \frac{K_2}{2})$$
$$K_4 = \Delta x f(x_j + \Delta x, y_j + K_3)$$

3 Runge-Kutta Method (Systems of Differential Equations)

These formulas may be modified for a system of ordinary differential equations by applying the formulas to each equation in the system, with each f_i as a function of x and the unknown functions $y^{(1)} \ldots y^{(n)}$. This method yields the following general equations.

Second-Order

$$y_{j+1}^{(1)} = y_j^{(1)} + \Delta x f_i\left(x_j + \frac{\Delta x}{2}, y_j^{(1)} + K_{1,1}, \ldots y_j^{(n)} + K_{1,n}\right)$$

where $K_{1,1} = \frac{\Delta x}{2} f_i(x_j, y_j^{(1)}, \ldots, y_j^{(n)})$

Fourth-Order

$$y_{j+1}^{(1)} = y_j^{(1)} + 1/6\,(K_{1,1} + 2K_{2,1} + 2K_{3,1} + K_{4,1})$$

where

$$K_{1,1} = \Delta x f_i(x_j, y_j^{(1)}, \ldots, y_j^{(n)})$$

$$K_{2,1} = \Delta x f_i\left(x_j + \frac{\Delta x}{2}, y_j^{(1)} + \frac{K_{1,1}}{2}, \ldots, y_j^{(n)} + \frac{K_{1,n}}{2}\right)$$

$$K_{3,1} = \Delta x f_i\left(x_j + \frac{\Delta x}{2}, y_j^{(1)} + \frac{K_{2,1}}{2}, \ldots, y_j^{(n)} + \frac{K_{2,n}}{2}\right)$$

$$K_{4,1} = \Delta x f_i(x_j + \Delta x, y_j^{(1)} + K_{3,1}, \ldots, y_j^{(n)} + K_{3,n})$$

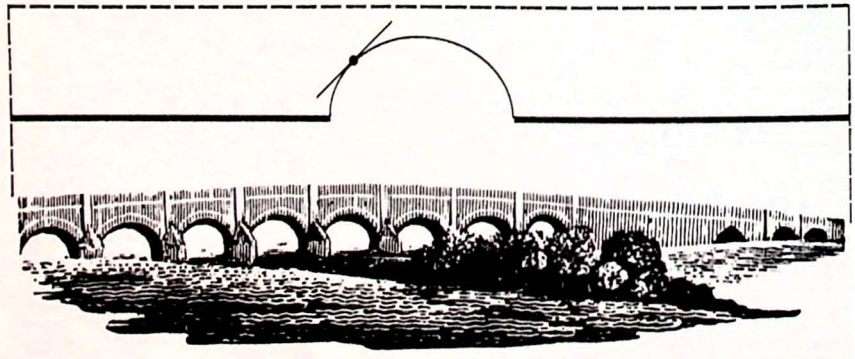
Note that the second-order equation may be written as

$$y_{j+1}^{(1)} = y_j^{(1)} + K_{2,1}$$

where $K_{2,1} = \Delta x f_i\left(x_j + \frac{\Delta x}{2}, y_j^{(j)} + \frac{K_{1,1}}{2}, \ldots, y_j^{(n)} + \frac{K_{1,n}}{2}\right)$

where $K_{1,1} = \Delta x f_i(x_j, y_j^{(1)}, \ldots, y_j^{(n)})$

The following program will solve a system of one or two differential equations. To use the program on the TI Programmable 58 you must repartition the calculator (Press 2 [2nd] [Up] 17).



THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 0 R/S STO 4 R/S STO 5 R/S STO 6 R/S	Stores Δx, x_0, $y_0^{(1)}$, and $y_0^{(2)}$ (if necessary).
014	2nd Lbl B RCL 4 STO 1 RCL 5 STO 2 RCL 6 STO 03	Moves j+1 values of x and y's for new integration step.
028	2nd Lbl C RCL 1 STO 7 RCL 2 STO 8 RCL 3 STO 9 2nd A'	Stores j values of x and y's to begin computation. Calls subroutine A' to compute f_1 and f_2.
043	RCL 10 × RCL 0 = STO 12 ÷ 2 + RCL 2 = STO 8 2nd If flg 1 079	Computes $K_{1,1}$ and stores it in R_{12}. Computes $y_j^{(1)} + \frac{K_{1,1}}{2}$ and stores it. If flag 1 is set, transfers to step #079. (Skips computation of $K_{1,2}$.)
063	RCL 11 × RCL 0 = STO 13 ÷ 2 + RCL 3 = STO 09	Computes $K_{1,2}$ and stores it in R_{13}. Computes $j_j^{(2)} + \frac{K_{1,2}}{2}$ and stores it.
079	RCL 1 + RCL 0 ÷ 2 = STO 7 2nd A'	Computes $x_j + \frac{\Delta x}{2}$ and stores it. Calls subroutine A' to compute f_1 and f_2.
090	RCL 10 × RCL 0 = STO 14 2nd If flg 1 110	Computes $K_{2,1}$ and stores it in R_{14}. If flag 1 is set, transfers to step #110. (Skips computation of $K_{2,2}$.)
102	RCL 11 × RCL 0 = STO 15	Computes $K_{2,2}$ and stores it in R_{15}.
110	2nd If flg 2 E RCL 14 ÷ 2 + RCL 2 = STO 8 2nd If flg 1 137	If flag 2 is set, transfers to subroutine E. Computes $y_j^{(1)} + \frac{K_{2,1}}{2}$ and stores it. If flag 1 is set, transfers to step #137.
127	RCL 15 ÷ 2 + RCL 3 = STO 09	Computes $y_j^{(2)} + \frac{K_{2,2}}{2}$

3

Runge-Kutta Method (Systems of Differential Equations)

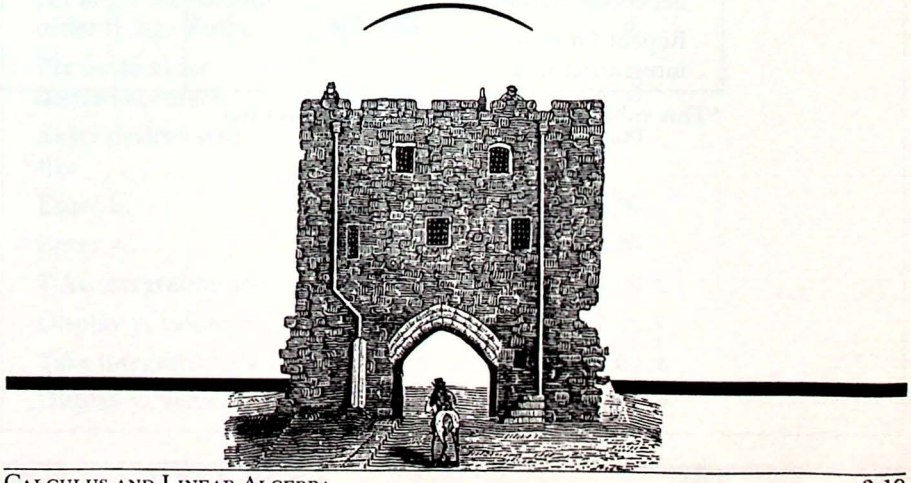
Step #	Program Keystrokes	Purpose/Comments
137	2nd A' RCL 10 × RCL 0 = STO 16 + RCL 2 = STO 8 2nd If flg 1 170	Calls subroutine A' to compute f_1 and f_2. Computes $K_{3,1}$ and stores it in R_{16}. Computes $y_j^{(1)} + K_{3,1}$ and stores it. If flag 1 is set, transfers to step #170.
156	RCL 11 × RCL 0 = STO 17 + RCL 3 = STO 09	Computes $K_{3,2}$ and stores it in R_{17}. Computes $y_j^{(2)} + K_{3,2}$ and stores it.
170	RCL 1 + RCL 0 = STO 07	Computes $x_j + \Delta x$.
178	2nd A' RCL 10 × RCL 0 = STO 18 + 2 × RCL 16 + 2 × RCL 14 + RCL 12 = ÷ 6 + RCL 2 = STO 5 2nd If flg 1 243	Calls subroutine A' to compute f_1 and f_2. Computes $K_{4,1}$ and stores it in R_{18}. Computes $y_{j+1}^{(1)}$. If flag 1 is set, transfers to step #243.
213	RCL 11 × RCL 0 = STO 19 + 2 × RCL 17 + 2 × RCL 15 + RCL 13 = ÷ 6 + RCL 3 = STO 06	Computes $K_{4,2}$ and stores it in R_{19}. Computes $y_{j+1}^{(2)}$.
243	RCL 7 STO 4 2nd Prt R/S	Stores $x_{j+1} = x_j + \Delta x$ in R_4 and displays x_{j+1}.
249	2nd Lbl D RCL 5 2nd Prt R/S RCL 6 2nd Prt R/S	Recalls and displays $y_{j+1}^{(1)}$ and $y_{j+1}^{(2)}$.
259	2nd Lbl E RCL 14 + RCL 2 = STO 5 2nd If flg 1 281	Computes $y_{j+1}^{(1)} = y_j^{(1)} + K_{2,1}$ for second-order Runge-Kutta. If flag 1 is set, transfers to step #281.
273	RCL 15 + RCL 3 = STO 06	Computes $y_{j+1}^{(2)} = y_j^{(2)} + K_{2,2}$.

Step #	Program Keystrokes	Purpose/Comments
281	RCL 1 + RCL 0 = STO 4 2nd Prt R/S GTO D	Computes $x_{j+1} = x_j + \Delta x$ for second-order Runge-Kutta. Transfers to subroutine D to display y_{j+1} values.
293	2nd Lbl 2nd A'	Label for function subroutine.
	LRN RST	Exits learn mode.

Note: Print commands may be replaced by Nop if the PC-100A is not used.

REGISTER CONTENTS

R_{00}	Δx	R_{07}	x	R_{14}	$K_{2,1}$
R_{01}	x_j	R_{08}	$y^{(1)}$	R_{15}	$K_{2,2}$
R_{02}	$y_j^{(1)}$	R_{09}	$y^{(2)}$	R_{16}	$K_{3,1}$
R_{03}	$y_j^{(2)}$	R_{10}	f_1	R_{17}	$K_{3,2}$
R_{04}	x_{j+1}	R_{11}	f_2	R_{18}	$K_{4,1}$
R_{05}	$y_{j+1}^{(1)}$	R_{12}	$K_{1,1}$	R_{19}	$K_{4,2}$
R_{06}	$y_{j+1}^{(2)}$	R_{13}	$K_{1,2}$		



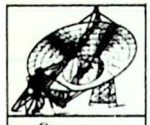
3

Runge-Kutta Method (Systems of Differential Equations)

Using the Program

Procedure	Press	Display/Comments
Enter subroutine for f_1 and f_2. Use the x and y values in registers 7, 8, and 9 when you need them.	GTO 2nd A'	
	LRN	295 00
	Keystrokes for f_1 and f_2.	
	INV SBR	
	LRN	0.
If you have only one equation, set flag 1.	2nd St flg 1	0.
If you wish to use a second-order Runge-Kutta, set flag 2.	2nd St flg 2	0.
Enter desired step size (Δx).	A	Δx
Enter x_0.	R/S	x_0
Enter $y_0^{(1)}$.	R/S	$y_0^{(1)}$
Enter $y_0^{(2)}$ (if necessary).	R/S	$y_0^{(2)}$
Take integration step.	B	x_{j+1}*
Display $y_{j+1}^{(1)}$. (Optional)	R/S	$y_{j+1}^{(1)}$*
Display $y_{j+1}^{(2)}$ if necessary. (Optional)	R/S	$y_{j+1}^{(2)}$*
Repeat for new integration step.		

*This value is printed if the PC-100A is used.



SAMPLE PROBLEM

(1) Newton's law of cooling states that the temperature of a body changes at a rate that is proportional to the difference in temperature between the body and the outside medium.

Suppose a thermometer is moved outside, where the air temperature is $-10°C$, from inside a house where the thermometer is reading 24°C. Since the time rate of change, dy/dt, is proportional to the temperature difference, $(y-(-10))$, we have the differential equation

$$\frac{dy}{dt} = -k\,(y+10)$$

Assume the constant of proportionality, k, is equal to 0.23 and use the second-order Runge-Kutta formula to solve the differential equation for $t=0$ to $t=5$ with a step size of 1.

Procedure	Press	Display/Comments
Enter program.		0
Enter subroutine for f_1.	GTO 2nd A	0.
	LRN	295 00
	.23 +/- × (	
	RCL 8 + 10)	
	= STO 10	
	INV SBR	311 00
	LRN	0.
Set flag 1 to indicate one equation.	2nd St flg 1	0.
Set flag 2 for second-order Runge-Kutta.	2nd St flg 2	0.
Fix decimal for desired accuracy.	2nd Fix 2	0.00
Enter desired step size.	1 A	1.00
Enter t_0.	0 R/S	0.00
Enter y_0.	24 R/S	24.00
Take integration step.	B	1.00 t_1
Display y_1 value.	R/S	17.08 y_1
Take integration step.	B	2.00 t_2
Display y_2 value.	R/S	11.57 y_2

Procedure	Press	Display/Comments
Continue until all values have been computed.	B	3.00 t_3
	R/S	7.18 y_3
	B	4.00 t_4
	R/S	3.68 y_4
	B	5.00 t_5
	R/S	0.90 y_5

(2) Consider a point mass leaving a plane with an initial velocity of zero. The body has two forces acting on it. First the body is pulled downward by gravity with a force of mg. However, there is also an upward force on the body which we can assume is proportional to its speed squared. Thus the total force acting on the body is:

$$F = mg - kv^2$$

where k is the constant that describes the damping force. To find the distance the body has fallen at any time, t, we must solve the second-order differential equation

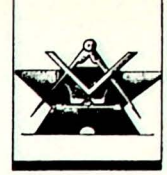
$$\frac{d^2y}{dt^2} = g - \frac{k}{m}\left(\frac{dy}{dt}\right)^2$$

We can use this program to numerically solve the equation for y by rewriting it as a system of two equations as follows:

$$f_1 = \frac{dy}{dt} = v$$

$$f_2 = \frac{dv}{dt} = g - \frac{k}{m}v^2$$

Solve this system of equations using fourth-order Runge-Kutta and a step size of 1 for $t = 0$ to $t = 10$. Assume $k = 0.005$, $m = 1$, and $g = 9.8$. Note that $t_0 = y_0 = v_0 = 0$. Compare your results with those obtained in the Euler's Method problem in the Physics chapter.

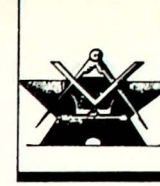


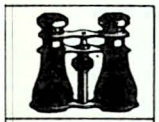
Procedure	Press	Display/Comments
Unset all flags and clear the memories.	RST CLR 2nd CMs	0 0.00
Enter subroutine for f_1 and f_2.	GTO 2nd A' LRN RCL 9 STO 10 (9.8 − .005 × RCL 9 x^2) STO 11 INV SBR LRN	 0.00
Enter desired step size.	1 A	1.00
Enter t_0.	0 R/S	0.00
Enter y_0.	0 R/S	0.00
Enter v_0.	0 R/S	0.00
Take integration step.	B	1.00 t_1
Display y_1.	R/S	4.86 y_1
Display v_1	R/S	9.64 v_1
Take next integration step.	B R/S R/S	2.00 t_2 18.99 y_2 18.41 v_2
Continue until all values are generated.	. . .	
	B R/S R/S	10.00 t_{10} 306.46 y_{10} 43.23 v_{10}

The following values will be generated:

1.00	t_1	6.00	t_6
4.86	y_1	140.57	y_6
9.64	v_1	38.46	v_6
2.00	t_2	7.00	t_7
18.99	y_2	180.09	y_7
18.41	v_2	40.45	v_7
3.00	t_3	8.00	t_8
41.19	y_3	221.25	y_8
25.72	v_3	41.78	v_8
4.00	t_4	9.00	t_9
69.89	y_4	263.50	y_9
31.39	v_4	42.65	v_9
5.00	t_5	10.00	t_{10}
103.48	y_5	306.46	y_{10}
35.55	v_5	43.23	v_{10}

These results for the y-values are better than those obtained by Euler's Method using the same step size. To get the same accuracy with Euler's Method you would have to use a much smaller step size, say around 0.01. You might note that since Euler's Method gives the speed (v) at the midpoint of the interval, you cannot readily compare the speed values unless you choose a step size that would give you the values at the same points.





FURTHER EXPLORATIONS

1. Solve the second-order differential equation

$$y'' - 2y' + y = 0 \qquad y(0) = 1,\ y'(0) = 1$$

by writing the equations as a system.

$$\frac{dy}{dx} = u \qquad y(0) = 1$$

$$\frac{du}{dx} = 2u - y \qquad u(0) = 1$$

Solve using $\Delta x = 1$ for values of y at $x = 1, 2, 3, 4, 5$. Compare the results with the actual solution

$$y(x) = e^x$$

2. Find the solutions to the system

$$\frac{dy}{dx} = y + 3u \qquad y(0) = 2$$

$$\frac{du}{dx} = 3y + u \qquad u(0) = 0$$

at $x = 0.5$ using $\Delta x = 0.5, 0.25, 0.1$, and 0.05 with the second-order method. Compare your results to the actual values

$$y(.5) = e^2 + e^{-1} = 7.757$$
$$u(.5) = e^2 - e^{-1} = 7.021$$

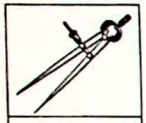
3. Using the same methods employed to solve systems of one or two equations, write a program which will solve a system of three differential equations.

REFERENCES/ INFORMATION

Gerald B. Haggerty, *Elementary Numerical Analysis with Programming* (Boston: Allyn and Bacon, Inc. 1972), pp. 388-406.

S. D. Conte and Carl de Boor, *Elementary Numerical Analysis,* 2nd ed. (New York: McGraw-Hill Book Company, 1972), pp. 336-340.

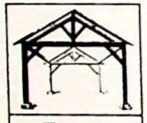
3 Curve-Fitting with Least-Squares Approximation



OBJECTIVE

When interpreting experimental data, a primary task is to find a function which will fit the discrete experimental results. Such a function exists, since the results of any continuous experiment are related to each other in some way, no matter how unusual or obscure the relationship may be. Because the exact function defining the relationship may be almost impossible to find, we usually look for an approximate function, one which will produce data close to the set of experimental data points and can be used to approximate the whole range of results within certain error bounds.

The class of approximating functions must be chosen with care, and the specific function from this class must be computed carefully. The program listed here calculates a system of equations that can then be reduced to yield a polynomial of chosen degree.



FACTS/ THEORY

Given a set of data points $[(x_1,y_1),(x_2,y_2) \ldots (x_n,y_n)]$, the idea is to find an approximate polynomial of degree m in the form

$$P(x) = a_0 + a_1x + a_2x^2 + \ldots + a_mx^m$$

where the errors $[E_i = y_i - P(x_i),\ i = 1 \ldots n]$ are small. If $n \leq m+1$, there is an exact fit, and this is just an interpolation problem. However, if $n > m+1$, the errors (E_i) are usually nonzero. In least-squares approximation the sum of the squares of the errors is made as small as possible.

Consider the function

$$f(a_0, \ldots a_m) = \sum_{i=1}^{n} (y_i - \sum_{j=1}^{m} a_j x_i^{\,j})^2$$

Using calculus to take derivatives of each a_j and to set them equal to zero (to find the extreme points), we have

$$\frac{\partial f}{\partial a_k} = -2 \sum_{i=1}^{n} \left[(y_i - \sum_{j=0}^{m} a_j x_i^{\,j}) x_i^{\,k} \right] = 0$$

for $k = 0, 1, 2, \ldots m$. This expression can be rewritten in two forms,

$$\sum_{i=1}^{n} y_i x_i^{\,k} - \sum_{i=1}^{m} \left(\sum_{j=0}^{m} a_j x_i^{\,k} \right) x_i^{\,k} = 0$$

or (collecting coefficients for each a_j)

$$\sum_{i=1}^{n} y_i x_i^{\,k} - \sum_{j=0}^{m} \left(\sum_{i=1}^{n} x_i^{\,j+k} \right) a_j = 0$$

This is a set of $m+1$ linear equations in the $m+1$ unknowns $a_0, a_1, \ldots a_m$. The set then can be solved by standard techniques, and the resulting polynomial is the least-squares approximation to the given data set.

The question of selecting the appropriate degree of the approximating polynomial is not easy to answer. One method is to plot the data points on a graph and sketch a polynomial that comes close to the data points. The number of turns or "humps" in the sketch is one less than the degree of the suggested polynomial. (For other analytical methods, see page 281 in the first referenced volume at the end of this section.)

The program shown below can be used to produce the linear system of equations

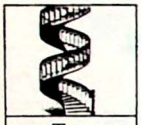
$$na_0 + \left(\sum_{i=1}^{n} x_i\right)a_1 + \ldots + \left(\sum_{i=1}^{n} x_i^m\right)a_m = \sum_{i=1}^{n} y_i$$

$$\left(\sum_{i=1}^{n} x_i\right)a_0 + \left(\sum_{i=1}^{n} x_i^2\right)a_1 + \ldots + \left(\sum_{i=1}^{n} x_i^{m+1}\right)a_m = \sum_{i=1}^{n} x_i y_i$$

$$\left(\sum_{i=1}^{n} x_i^m\right)a_0 + \left(\sum_{i=1}^{n} x_i^{m+1}\right)a_1 + \ldots + \left(\sum_{i=1}^{n} x_i^{2m}\right)a_m = \sum_{i=1}^{n} x_i^m y_i$$

for the $a_0, a_1 \ldots a_m$ coefficients. Another program, like the row-reduction program listed later in this chapter or ML-02 in the Master Library, can then be used to solve this system.

The program requires the use of $3m+2n+10$ memory registers for data storage. Be sure to repartition the data/program registers appropriately if you need additional memories for the problem you are evaluating. Also, the program operates only if x values ≥ 0 are used. If negative x values are present in your data set, you should adjust the program to calculate x^k without creating an error condition.



THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 1 × 3 + 10 = STO 3 STO 4 R/S	Stores m and computes and stores the highest number of registers needed to store the sums for the linear equations $(3m + 10)$.
015	2nd Lbl B STO 2 R/S	Stores n (number of data points).
020	2nd Lbl C 2nd Op 23 STO 2nd Ind 3 R/S	Stores $x_1, \ldots, x_n$ starting in register $3m + 11$.
027	2nd Lbl D 2nd Op 23 STO 2nd Ind 3 R/S	Stores $y_1, \ldots, y_n$.
034	2nd Lbl E 0 STO 5 STO 06 9 STO 8 RCL 1 × 2 = x⇄t	Initializes registers for computation of the sums $\sum_{i=1}^{n} x_i^k$.
050	2nd Lbl 2nd A' RCL 2 STO 0 RCL 4 STO 3 2nd Op 25 CLR STO 07	Stores 2m in the t-register for testing later. Note that subroutine A' is run for every value of k.
065	2nd Lbl 2nd B' 2nd Op 23 RCL 2nd Ind 3 y^x RCL 5 = SUM 7 2nd Dsz 0 2nd B'	Computes $\sum_{i=1}^{n} x_i^k$ for $k = 1, 2, \ldots, 2m$.
080	2nd Op 28 RCL 7 STO 2nd Ind 8 RCL 5 INV 2nd x≠t 2nd A'	Stores $\sum_{i=1}^{n} x_i^k$ beginning in register 10. If $k < 2m$, transfers to A' to reinitialize counters for new k.

Step #	Program Keystrokes	Purpose/Comments
091	2nd Lbl 2nd C' RCL 3 STO 9 RCL 2 STO 0 RCL 4 STO 3 CLR STO 07	Initializes registers for computation of the sums $\sum_{i=1}^{n} y_i x_i^k$
108	2nd Lbl 2nd D' 2nd Op 23 2nd Op 29 RCL 2nd Ind 3 y^x RCL 6 × RCL 2nd Ind 9 = SUM 7 2nd Dsz 0 2nd D'	Computes $\sum_{i=1}^{n} y_i x_i^k$ for k = 0, 1, . . . , m.
128	2nd Op 28 RCL 7 STO 2nd Ind 8 RCL 1 $x \rightleftarrows t$ RCL 6 2nd $x=t$ 2nd E' 2nd Op 26 GTO 2nd C'	Stores $\sum_{i=1}^{n} y_i x_i^k$. If k≥m, transfers to E' to recall results. if k<m, transfers to C' for new k.
145	2nd Lbl 2nd E' 2nd Prd 02 9 STO 0 2nd Lbl 2nd List 2nd Op 20 RCL 2nd Ind 0 2nd Prt R/S 2nd Dsz 2 2nd List CLR GTO CLR	Initializes registers for recalling sums computed. Recalls and displays $\sum_{i=1}^{n} x_i^k$ for k = 1, 2, . . . , 2m. Then recalls and displays $\sum_{i=1}^{n} y_i x_i^k$ for k = 0, 1, 2, . . . , m. When all sums have been displayed, a flashing zero shows in the display.
	LRN RST	Exits learn mode.

Note: Print command may be omitted if the PC-100A is not used.

REGISTER CONTENTS

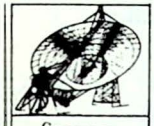
R_{00}	Used	R_{05}	k	R_{10}	(Remaining registers used
R_{01}	m	R_{06}	k	R_{11}	for summations and data
R_{02}	n, 2n	R_{07}	Used	R_{12}	storage, depending on
R_{03}	Used	R_{08}	Used	R_{13}	value of m.)
R_{04}	3m + 10	R_{09}	Used	R_{14}	

USING THE PROGRAM

Procedure	Press	Display / Comments
Enter degree of polynomial, m.	A	3m + 10
Enter number of data points, n.	B	n
Enter x_i data.	x_1 C	x_1
	x_2 C	x_2
	• • •	• • •
	x_n C	x_n
Enter y_i data.	y_1 D	y_1
	y_2 D	y_2
	• • •	• • •
	y_n D	y_n
Calculate coefficients (C) and constants (K).	E	C_1° Σx_i
	R/S	C_2° Σx_i^2
	R/S	C_3° Σx_i^3
	• • •	• • •
	R/S	C_{2m}° Σx_i^{2m}
	R/S	K_1° $\Sigma y_i x_i^0$
	R/S	K_2° $\Sigma y_i x_i^1$
	• • •	• • •
	R/S	K_{m+1}° $\Sigma y_i x_i^m$
	R/S	0 Flashing; indicates end of run.

°This value is printed if the PC-100A is used.

Note: Since the coefficients are repeated diagonally in the linear set of equations, each coefficient is calculated only once, in the order of first appearance.



SAMPLE PROBLEM

Find a least-squares approximation for the following data set.

x	0	1	1.5	3	4.5
y	−2	.04	2.91	−1.6	1.1

Plotting these points indicates that a cubic polynomial will give good results; therefore, let m = 3.

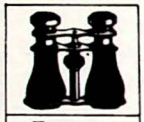
Procedure	Press	Display/ Comments
Enter program.		0
Clear memories before each program run.	2nd CMs	0.
Enter m.	3 A	19.
Enter n.	5 B	5.
Enter x_i data.	0 C	0.
	1 C	1.
	1.5 C	1.5
	3 C	3.
	4.5 C	4.5
Enter y_i data.	2 +/− D	−2.
	.04 D	0.04
	2.91 D	2.91
	1.6 +/− D	−1.6
	1.1 D	1.1
Calculate coefficients and constants for linear equations.	E	10. Σx_i
	R/S	32.5 Σx_i^2
	R/S	122.5 Σx_i^3
	R/S	497.125 Σx_i^4
	R/S	2096.875 Σx_i^5
	R/S	9045.15625 Σx_i^6
	R/S	0.45 $\Sigma y_i x_i^0 = \Sigma y_i$
	R/S	4.555 $\Sigma y_i x_i^1$
	R/S	14.4625 $\Sigma y_i x_i^2$
	R/S	66.89875 $\Sigma y_i x_i^m$
	R/S	0 Flashing—all values generated.

Putting these values into their correct places gives this linear system of equations.

$$\begin{aligned}
5a_0 + 10a_1 + 32.5a_2 + 122.5a_3 &= 0.45 \\
10a_0 + 32.5a_1 + 122.5a_2 + 497.125a_3 &= 4.555 \\
32.5a_0 + 122.5a_1 + 497.125a_2 + 2096.875a_3 &= 14.4625 \\
122.5a_0 + 497.125a_1 + 2096.875a_2 + 9045.15625a_3 &= 66.89875
\end{aligned}$$

You can now solve the system by one of the methods discussed earlier. The resulting polynomial (rounded to four places) is

$$P(x) = -2.2464 + 7.0674x - 3.9112x^2 + 0.5561x^3$$



FURTHER EXPLORATIONS

Use the program to compute the least-squares approximation to the data set

x	0	1	2	3	4
y	1	2.14	−1.6	−7.1	6.2

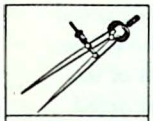
using polynomials of degree 1, 2, and 3. Compute the error for each polynomial.

REFERENCES/ INFORMATION

Gerald B. Haggerty, *Elementary Numerical Analysis with Programming* (Boston: Allyn and Bacon, 1972), pp. 272-295.

F. B. Hildebrand, *Introduction to Numerical Analysis,* 2nd ed. (New York: McGraw-Hill Book Company, 1974), pp. 314-326.

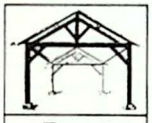
Systems of Linear Equations (Row Reduction)



OBJECTIVE

There are several methods of solving simultaneous equations. The Master Library program ML-02 (see "Systems of Linear Equations" in the College Algebra and Trigonometry chapter) can be used to solve simultaneous equations, as well as to evaluate determinants and calculate the inverse of the coefficient matrix. The method shown here is row reduction, and this program is more efficient than the Master Library program for the specific task of solving linear equations.

Because of the length of the program and the memory storage required, this program is restricted to the TI Programmable 59. Owners of the TI Programmable 58 should refer to the Master Library program mentioned above.



FACTS/ THEORY

Consider the following representation of a series of simultaneous equations.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ &\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n \end{aligned}$$

The coefficients can be assembled into the following matrix notation.

$$[A,b] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{bmatrix}$$

Three basic rules apply to the manipulation of a series of equations that naturally relate to the matrix representation of those equations.

1. Because the given equations can be listed in any order, any two rows of a matrix can be interchanged.
2. Both sides of an equation can be multiplied by some constant without changing the solution. Likewise, all elements in a row of a matrix can be multiplied by some constant without affecting the end result.
3. Any two equations can be added together, as can two rows of a matrix.

When applied to equations, this means that the coefficients of one equation can be multiplied by a constant that will make the coefficient of one of the unknowns equal in magnitude but opposite in sign to the coefficient of the same unknown in another equation. When the two equations are summed, that unknown is cancelled out, leaving a simplified equation with one less unknown – the equation has been reduced by one unknown.

Consider the following series of equations.

$$\begin{aligned} x+2y&=3\\ 3x+4y&=1 \end{aligned}$$

When the first equation is multiplied by -3 and the resulting equation added to the second, we get:

$$\begin{aligned} -3x-6y&=-9\\ 3x+4y&=1\\ \hline -2y&=-8\\ y&=4 \end{aligned}$$

This same method of reduction can be applied to matrices.

1. Divide each element of the first row of a matrix by the left-most element, a_{11}. If a_{11} is zero, simply exchange the whole top row with some other row whose left-most digit is nonzero. Now perform the above operation. This makes a_{11} equal to 1.
2. Add multiples of the first row to each other row so that the first (left-most) element of each row is zero. For instance, if the first element of row 2, a_{21}, is 3, multiply each element of row 1 by (-3) and add them respectively to row 2, making $a_{21}=0$. If a_{31} is -2, multiply row 1 by $(+2)$ and add it to the elements of row 3, etc. Continue this process until the matrix has the following form.

$$\begin{bmatrix} 1 & a'_{12} & a'_{13} & \cdots & a'_{1n} & b'_1 \\ 0 & a'_{22} & a'_{23} & \cdots & a'_{2n} & b'_2 \\ 0 & a'_{32} & a'_{33} & \cdots & a'_{3n} & b'_3 \\ \downarrow & \downarrow & \downarrow & \cdots & \downarrow & \downarrow \\ 0 & a'_{n2} & a'_{n3} & \cdots & a'_{nn} & b'_n \end{bmatrix}$$

3. Now ignore the first row and column and consider the resulting submatrix of $n-1$ rows. Proceed to reduce this new matrix just as we did the original matrix. Continue for each new submatrix until the following form is obtained.

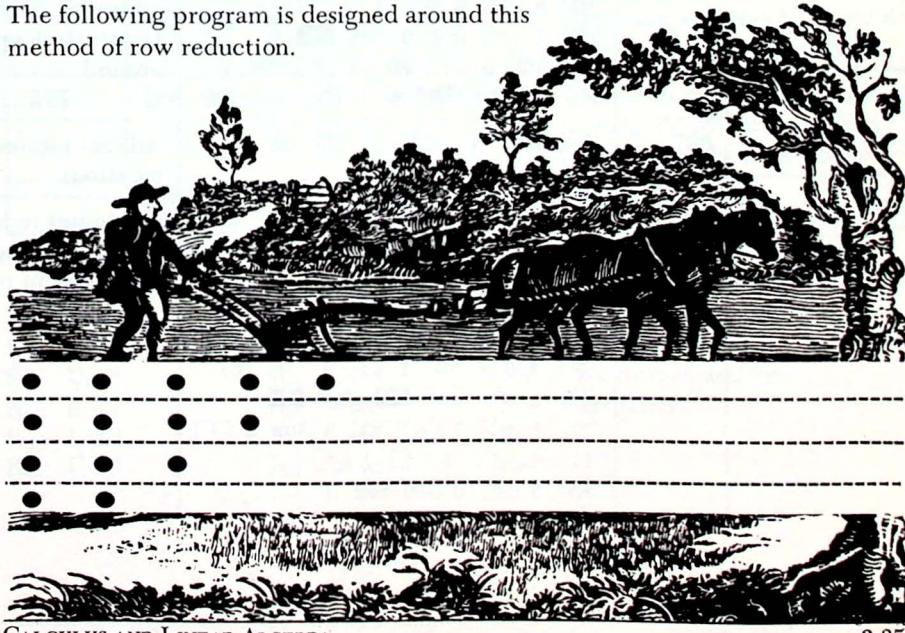
$$\begin{bmatrix} 1 & a'_{12} & a'_{13} & \cdots & a'_{1n} & b'_1 \\ 0 & 1 & a'_{23} & \cdots & a'_{2n} & b'_2 \\ 0 & 0 & 1 & \cdots & a'_{3n} & b'_3 \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 1 & b'_n \end{bmatrix}$$

These represent the coefficients of a new series of simultaneous equations.

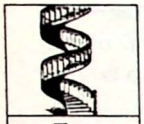
$$\begin{aligned} x_1 + a'_{12}x_2 + a'_{13}x_3 + \longrightarrow a'_{1n}x_n &= b'_1 \\ x_2 + a'_{23}x_3 + \longrightarrow a'_{2n}x_n &= b'_2 \\ x_3 + \longrightarrow a'_{3n}x_n &= b'_3 \\ \cdot \quad & \cdot \\ \cdot \quad & \cdot \\ \cdot \quad & \cdot \\ x_n &= b'_n \end{aligned}$$

Knowing that $x_n = b'_n$ and substituting backward can eventually yield all the values from x_n through x_1.

The following program is designed around this method of row reduction.

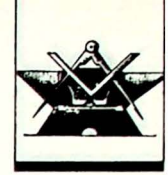


Systems of Linear Equations (Row Reduction)



The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [A] [2nd] [CMs] [STO] 1 [STO] 0 [+] 1 [=] [STO] 03	Stores number of equations (n) and number of columns (n + 1).
012	12 [STO] 2 [STO] 8 [2nd] [Op] 38 [CLR] [R/S]	Enters 12 for indirect memory storage of later data entries.
022	[2nd] [Lbl] [B] [STO] [2nd] [Ind] 2 [2nd] [Op] 22 [R/S]	Stores data (a_{11} stored in memory 12, a_{12} stored in memory 13, etc.).
029	[2nd] [Lbl] [2nd] [A'] [RCL] 1 [−] 1 [−] [RCL] 4 [=] [STO] 9 [2nd] [Op] 28 [RCL] 8 [STO] 7 [RCL] [2nd] [Ind] 8 [STO] 2 [INV] [2nd] [St flg] 0	Begins calculation. Clears flag 0 for looping purposes later in program.
054	[2nd] [Lbl] [C] [RCL] 3 [SUM] 8 [RCL] [2nd] [Ind] 8 [÷] [RCL] 2 [=] [STO] 10 [RCL] 8 [STO] 5 [RCL] 3 [−] 1 [−] [RCL] 4 [=] [STO] 0 [2nd] [If flg] 0 [D] [RCL] 5 [−] [RCL] 3 [=] [STO] 6 [STO] 11 [2nd] [St flg] 0	Calculates reduction factor and adjusts indirect memory addresses. Sets flag 0 to terminate loop next time around.
097	[2nd] [Lbl] [D] [RCL] 11 [STO] 06	Adjusts indirect memory locations.
103	[2nd] [Lbl] [E] [2nd] [Op] 25 [2nd] [Op] 26 [RCL] [2nd] [Ind] 5 [−] [RCL] 10 [×] [RCL] [2nd] [Ind] 6 [=] [STO] [2nd] [Ind] 5 [2nd] [Dsz] 0 [E] [2nd] [Dsz] 9 [C] [2nd] [Op] 24 [RCL] 7 [+] [RCL] 3 [=] [STO] 8 [RCL] 1 [−] 1 [=] [x≥t] [RCL] 4 [INV] [2nd] [x≥t] [2nd] [A'] [RCL] 1 [STO] 7 [×] [RCL] 3 [STO] 4 [+] 11 [=] [STO] 6 [+] 1 [=] [STO] 5 [STO] 0 [2nd] [St flg] 1	Continues reduction of terms. Sets flag 1 to control loop later in program.



Step #	Program Keystrokes	Purpose/Comments
171	2nd Lbl 2nd E' RCL 6 STO 8 RCL 3 − RCL 4 = STO 2 RCL 2nd Ind 8 STO 10 2nd If flg 1 2nd B' RCL 0 STO 05	Adjusts indirect memory locations for loops and recall of correct data.
196	2nd Lbl 2nd C' 2nd Op 38 RCL 10 − RCL 2nd Ind 8 × RCL 2nd Ind 5 = STO 10 2nd Op 25 2nd Dsz 2 2nd C'	Continues reduction.
216	2nd Lbl 2nd B' INV 2nd St flg 1 RCL 8 − 1 = STO 9 RCL 2nd Ind 9 x⇄t 0 2nd x=t 2nd D' RCL 10 ÷ RCL 2nd Ind 9 = STO 2nd Ind 5 RCL 6 − RCL 3 = STO 6 2nd Op 34 2nd Dsz 7 2nd E'	Continues reduction and adjusts indirect memory locations.
255	2nd Lbl lnx RCL 2nd Ind 5 EE INV EE 2nd Prt R/S	Displays x_i and stops.
264	2nd Op 35 RCL 0 x⇄t RCL 5 2nd x=t lnx	Adjusts indirect memory location and checks to see if another loop is required.
273	2nd Lbl 2nd D' 1 +/− √x R/S	Displays flashing 1 and stops to signify end of run. All x_i terms have been generated.
	LRN RST	Exits learn mode.

Note: Print command may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}	Used	R_{06}	Used	R_{12}	(Remaining registers
R_{01}	n	R_{07}	Used	R_{13}	used as needed for
R_{02}	Used	R_{08}	Used	R_{14}	data storage and/or
R_{03}	n + 1	R_{09}	Used	R_{15}	summation.)
R_{04}	Used	R_{10}	Used	R_{16}	
R_{05}	Used	R_{11}	Used	R_{17}	

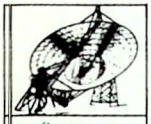
3 Systems of Linear Equations (Row Reduction)

Using the Program

Procedure	Press	Display/Comments
Enter number of equations, n (see note below).	A	0
Enter coefficients (a_{ij}) and constant (b_i) by rows. Continue for each equation.	a_{11} B a_{12} B • • • b_1 B a_{21} B a_{22} B • • • b_n B	a_{11} Data must be a_{12} entered in this • • • order. Data b_1 storage begins a_{21} in memory a_{22} register 12. • • • b_n
When all values have been entered, calculate x values. Continue to press R/S until all x values have been calculated.	2nd A' R/S • • • R/S R/S	x_1 ° x_2 ° ••• x_n ° 1. Flashing; indicates all x values have been generated.

°This value is printed if the PC-100A is used.

Note: As many as six simultaneous equations can be solved with the power-up partitioning. For more equations, repartition the program steps/memory registers as needed.

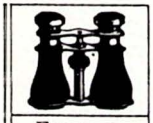


SAMPLE PROBLEM

Solve the three following simultaneous equations.

$$5x + 3y + 4z = 23$$
$$-2x + 3z = 7$$
$$3x - 2y + z = 2$$

Procedure	Press	Display/ Comments
Enter n.	3 [A]	0
Enter data by rows.		
a_{11}	5 [B]	5. a_{11}
a_{12}	3 [B]	3. a_{12}
a_{13}	4 [B]	4. a_{13}
b_1	23 [B]	23. b_1
a_{21}	2 [+/−] [B]	−2. a_{21}
a_{22}	0 [B]	0. a_{22}
a_{23}	3 [B]	3. a_{23}
b_2	7 [B]	7. b_2
a_{31}	3 [B]	3. a_{31}
a_{32}	2 [+/−] [B]	−2. a_{32}
a_{33}	1 [B]	1. a_{33}
b_3	2 [B]	2. b_3
Calculate x values.	[2nd] [A']	1. $x_1 = x$
	[R/S]	2. $x_2 = y$
	[R/S]	3. $x_3 = z$
	[R/S]	1. Flashing — no more x values.



FURTHER EXPLORATIONS

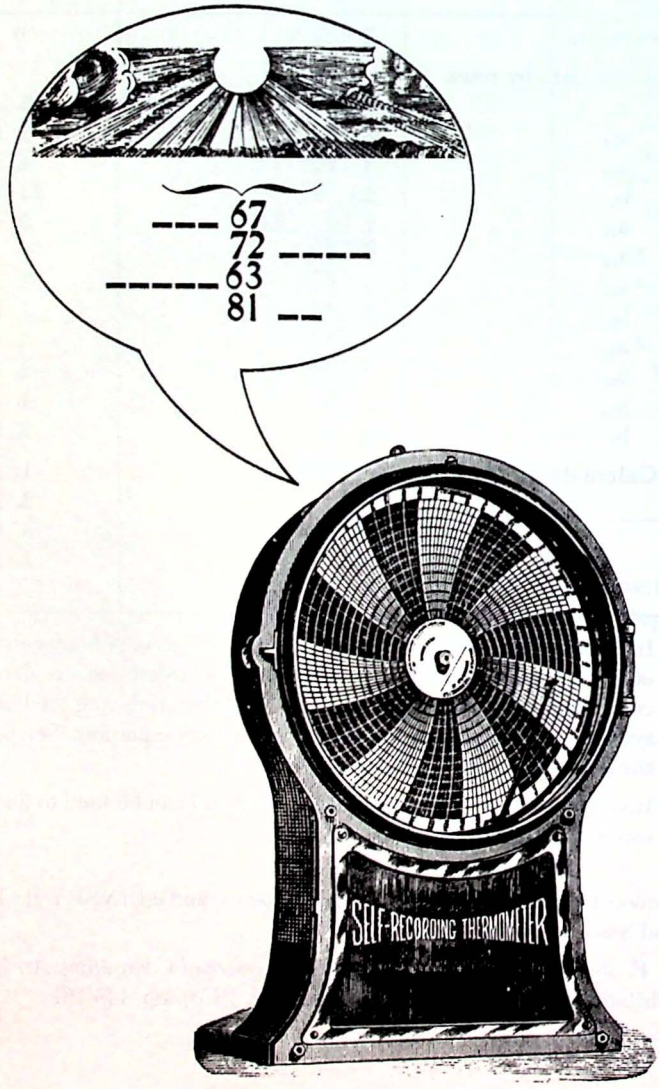
1. Investigate the solution procedure for a system of N homogeneous equations with N unknowns by finding the largest nonzero determinant contained in the coefficient array A, and then reducing the homogeneous system to a nonhomogeneous system of fewer equations. See pages 18-21 in the first reference listed below.
2. Investigate how the row reduction program can be used to find linearly independent sets of vectors.

REFERENCES/ INFORMATION

Howard Anton, *Elementary Linear Algebra,* 2nd ed. (New York: John Wiley and Sons, 1977), pp. 1-17.

L. F. Shampene and R. C. Allen, Jr., *Numerical Computing: An Introduction* (Philadelphia: W. B. Saunders Company, 1973), pp. 145-161.

4 Statistics



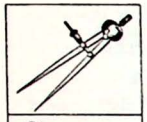
Introduction

Most of us are familiar with some of the general aspects of statistics. We read about local crime statistics in our newspapers; we follow with interest the "win-loss" statistics of our favorite sports teams; we hear about the probability of rain tomorrow, the predicted increase in the cost of living next year, and so on.

There are basically two types of statistics. One is *descriptive statistics*, which involves collecting, grouping, and presenting large sets of data in ways that can be easily understood or assimilated. The other kind, *inductive statistics* or *statistical inference*, is used to draw conclusions from your observed data — to estimate population parameters from sample data, for example, and/or to predict trends and explore probabilities.

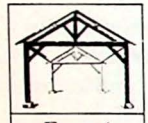
In all of these areas your programmable calculator can be an invaluable tool. The built-in statistical functions of the calculator and its programmability allow you to perform statistical calculations with ease and rapidity. You'll find examples in this chapter dealing with the descriptive measures of statistics (mean, standard deviation, variance, etc.), several kinds of distributions important in statistical analysis, linear regression, and the analysis of variance. These examples were selected to demonstrate the power and versatility of the programmable calculator in this field and to help you develop some of the techniques you need to explore the science of statistics even further on your own.

4 Descriptive Measures



OBJECTIVE

When you are describing a large set of data, a great deal of information about the data can be gained when the proper statistical techniques are applied. Some of the most commonly used descriptive measures are the mean, the variance, the standard deviation, the moments about zero, the moments about the mean, skewness, and kurtosis. This section explores these measures and develops a program for their calculation.



FACTS/ THEORY

Here's a set of numbers (they could be test scores, results of experiments, etc.):

3	4	1	8
6	7	1	4
8	5	3	3
2	2	6	5

Certain representative numbers, derived from the elements of the data set, may be used effectively to describe the set. The following paragraphs deal with these descriptive numbers (statistics) and how they are derived.

MOMENTS ABOUT ZERO

What happens if we

(1) raise each element in the set to some power,
(2) add the resulting numbers, and then
(3) divide the sum by the number of elements in the set?

The statistic obtained by using the first power in *step (1)* is the first moment about zero, also called the *mean*. The mean (or average) is a very important statistical measure, and your calculator has a built-in program to compute the mean for you, once your data are entered.

Statisticians use two symbols to represent the mean, depending upon the kind of data they are evaluating and the information they want to convey. For example, let's consider that the numbers given above are the scores made by an entire second-grade class on a particular test. If we want information about this particular class only, that is, the scores of the class make up the whole *population* we are evaluating, the symbol used for the mean is μ. If, however, we are looking for data about the performance of all the second-graders in the state's public school system, the 16 scores are just our sample, from which we will make certain inferences about the whole population. In this case the symbol used for the mean is $\bar{x}$.

If the second power is used in step (1), the second moment about zero is generated; if the third power is used, you obtain the third moment about zero, and so on.

The first moment about zero (the mean) is written as

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N}$$

where x_i = the elements of the set, and $\sum_{i=1}^{N} x_i$ = the sum of all the elements, $\bar{x}$ = the mean or average, and N = the number of elements. The second moment about zero is written

$$\frac{\sum_{i=1}^{N} x_i^2}{N}$$

where $\sum_{i=1}^{N} x_i^2 = x_1^2 + x_2^2 + x_3^2 + \ldots x_N^2$.

(When it is clear that all the elements of a set are to be summed, $\sum_{i=1}^{N} x_i$ is often written as Σx_i. We'll use this notation for simplicity from this point on.) The third and fourth moments about zero, then, are written

$$\frac{\Sigma x_i^3}{N} \text{ and } \frac{\Sigma x_i^4}{N}$$

MOMENTS ABOUT THE MEAN

Another group of statistical data is called the moments about the mean. These are derived by

(1) subtracting the mean ($\bar{x}$) from each element (x_i) of the set,
(2) raising each result to some power,
(3) adding the results, and
(4) dividing the sum by the number of elements in the set.

Thus the first moment about the mean is

$$\frac{\Sigma(x_i-\bar{x})}{N}$$

where $\Sigma(x_i-\bar{x}) = (x_1-\bar{x}) + (x_2-\bar{x}) \ldots + (x_N-\bar{x})$. The first moment about the mean is always equal to zero, since the differences obtained from elements greater than the mean are cancelled out by the differences obtained from elements less than the mean.

The second moment about the mean is another very important statistic called the variance (σ^2), and the square root of the variance is the standard deviation (σ) or standard error. Both of these tell you something about the "spread" or grouping of your data. The second moment about the mean is written

$$\sigma^2 = \text{VAR} = \frac{\Sigma(x_i-\bar{x})^2}{N}$$

and the standard deviation is

$$\sigma = \sqrt{\frac{\Sigma(x_i-\bar{x})^2}{N}}$$

The symbols given above for variance and standard deviation are those used to indicate that data from a whole population has been evaluated. When we work with a sample, however, other symbols are often used and other methods employed to give a better approximation of population statistics (parameters). In this chapter the following notation and methods are used for sample data:

$$\text{Variance} = s^2 = \frac{\Sigma(x_i-\bar{x})^2}{n-1}$$

$$\text{Sample standard deviation} = s = \sqrt{\frac{\Sigma(x_i-\bar{x})^2}{n-1}}$$

where n = the number of elements in the sample. The calculation of the sample standard deviation with n–1 weighting (dividing by n–1, rather than n) gives a better estimation of σ, the population standard deviation. Because the expected value of s is equal to σ, s is called an *unbiased estimate* of σ.

The variance and the standard deviation are included in the built-in statistics functions of your calculator, and you may use either N or n–1 weighting to compute these values.

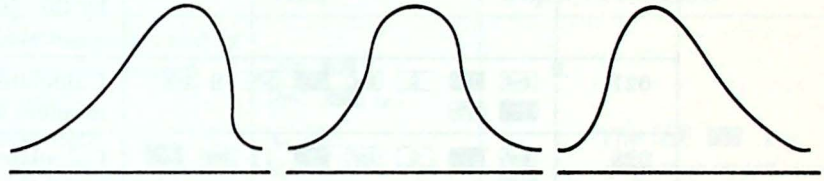
The third and fourth moments about the mean are expressed as

$$\frac{\Sigma(x_i-\bar{x})^3}{N} \quad \text{and} \quad \frac{\Sigma(x_i-\bar{x})^4}{N}$$

and these two values, along with the second moment about the mean, are used to calculate skewness and kurtosis.

SKEWNESS

Skewness is a measure of how symmetrical a curve your data would form if it were graphed:



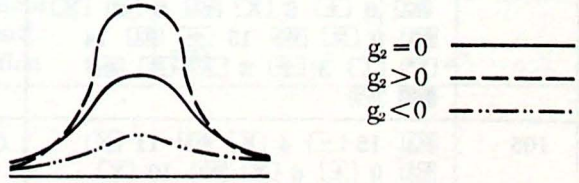
Negatively skewed (skewed to the left)	Not skewed	Positively skewed (skewed to the right)
$g_1 < 0$	$g_1 = 0$	$g_1 > 0$

The equation for determining skewness (g_1) is

$$g_1 = \frac{\Sigma(x_i-\bar{x})^3/N}{[\Sigma(x_i-\bar{x})^2/N]^{3/2}}$$

KURTOSIS

Kurtosis (g_2) is a measure that indicates the peakedness of a distribution.



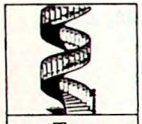
The expression for kurtosis is

$$g_2 = \frac{\Sigma(x_i-\bar{x})^4/N}{\Sigma(x_i-\bar{x})^2/N^2} - 3$$

4

Descriptive Measures

The program listed below incorporates the built-in functions for finding mean, variance, and standard deviation using either N or n–1 weighting. It then goes on to calculate skewness and kurtosis with N weighting.



The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 8 x^2 × RCL 8 = SUM 11 × RCL 8 = SUM 12 RCL 8 2nd $\Sigma+$ R/S	Enters x_i; calculates and sums x_i^3 and x_i^4 (x_i and x_i^2 are automatically summed by the 2nd $\Sigma+$ key sequence).
021	2nd Lbl B 2nd $\bar{x}$ STO 9 2nd Prt R/S	Calculates mean (1st moment about zero).
028	2nd Lbl C 2nd Op 11 2nd Prt R/S	Calculates σ^2 with N weighting.
034	$\sqrt{x}$ 2nd Prt R/S	Calculates σ with N weighting.
037	2nd Lbl 2nd C' INV 2nd $\bar{x}$ STO 8 x^2 2nd Prt R/S RCL 8 2nd Prt R/S	Calculates s and s^2 with n–1 weighting.
050	2nd Lbl D 2nd Op 11 STO 14 RCL 2 STO 10 RCL 3 INV 2nd Prd 10 INV 2nd Prd 11 INV 2nd Prd 12	Calculates and stores 2nd, 3rd, and 4th moments about zero.
071	RCL 11 − 3 × RCL 10 × RCL 9 + 2 × RCL 9 x^2 × RCL 9 = STO 15 ÷ RCL 14 y^x (3 ÷ 2) = 2nd Prt R/S	Calculates skewness.
105	RCL 12 − 4 × RCL 11 × RCL 9 + 6 × RCL 10 × RCL 9 x^2 − 3 × RCL 9 x^2 x^2 = STO 16 ÷ RCL 14 x^2 − 3 = 2nd Prt R/S	Calculates kurtosis.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS				
R_{00}		R_{09}	$\Sigma x_i/N$
R_{01}	Σx_i	R_{10}	$\Sigma x_i^2/N$
R_{02}	Σx_i^2	R_{11}	$\Sigma x_i^3/N$
R_{03}	N	R_{12}	$\Sigma x_i^4/N$
R_{04}	Used	R_{13}	
R_{05}	Used	R_{14}	VAR, $\Sigma(x_i-\bar{x})^2/N$
R_{06}	Used	R_{15}	$\Sigma(x_i-\bar{x})^3/N$
R_{07}		R_{16}	$\Sigma(x_i-\bar{x})^4/N$
R_{08}	Used	R_{17}	

USING THE PROGRAM

Procedure	Press	Display/Comments
Clear memories and fix decimal for desired accuracy.	2nd CMs 2nd Fix n	0.
Enter data.	x_1 A x_2 A x_3 A . . . x_N A	1. 2. 3. . . . N
When all x_i values have been entered, calculate mean.	B	$\bar{x}$° (or μ for population data.)
For population data (N weighting): *1.* Calculate variance.	C	σ^2°
2. Calculate standard deviation.	R/S	σ°

The 2nd Σ+ key sequence counts your data entries for you.

Procedure	Press	Display/Comments
OR		
For sample data (n–1 weighting):		
1. Calculate variance.	2nd C	s^2 *
2. Calculate standard deviation.	R/S	s *
Calculate skewness and kurtosis with N weighting.	D R/S	g_1 * Skewness g_2 * Kurtosis

*This value is printed if the PC-100A is used.

SAMPLE PROBLEM

Considering this set of numbers as sample data from a large population, calculate the mean, variance, standard deviation, skewness, and kurtosis:

3	4	1	8
6	7	1	4
8	5	3	3
2	2	6	5

Procedure	Press	Display/Comments
Enter program.		0
Clear memories and set decimal.	2nd CMs 2nd Fix 2	0. 0.00 We'll use two-place decimal for this example.
Enter data by columns.	3 A 6 A 8 A . . .	1.00 2.00 3.00 . . .
Continue to enter data, pressing A after each entry, until all elements have been entered.	5 A	16.00

Procedure	Press	Display/Comments
Calculate mean.	B	4.25 $\bar{x}$
Calculate sample variance and sample standard deviation.	2nd C'	5.27 s^2
	R/S	2.29 s
Calculate skewness and kurtosis.	D	0.23 g_1
	R/S	-1.07 g_2

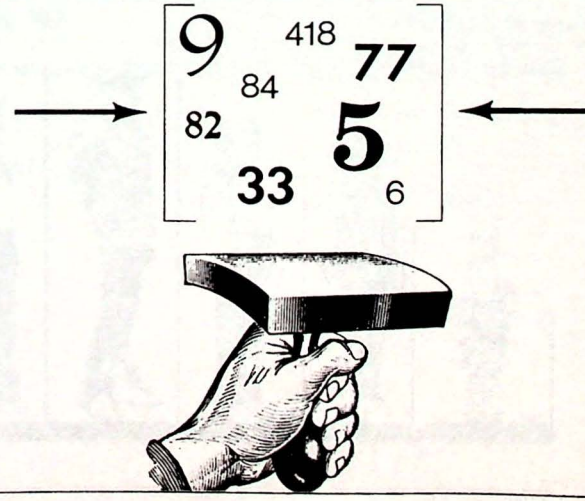
You now have quite a bit of statistical information about your data set. You know that the mean ($\bar{x}$) is 4.25 and the standard deviation (s) of your data is 2.29. Therefore, you can expect that roughly one-half of your data is grouped in the area of $\bar{x} \pm s$. That is, the range for about one-half of your elements is from 1.96 to 6.54.

Since skewness is positive, you know that the curve representing your data is slightly skewed to the right. The negative result for kurtosis means that the curve is less peaked than a normal curve (we'll discuss the normal curve in later sections).

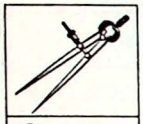
REFERENCES/ INFORMATION

Victor E. McGee, *Principles of Statistics: Traditional and Bayesian* (New York: Appleton-Century-Crofts, 1971), chapter 2.

For additional information about the statistical functions of your calculator, see *Personal Programming*, pp. V-32ff.



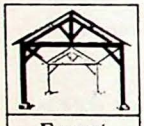
4 The Normal Curve



OBJECTIVE

Statisticians have found that the normal curve (normal distribution) approximates a vast number of real-life distributions. Also, even though the data for a population may not be normally distributed, calculations based on the normal curve can often be used if the sample size is large enough (generally, if $n \geq 30$). Sample data tend to arrange themselves normally in large samples even when the population is not normal.

Calculations of this sort have traditionally required the consultation of tables and standardized z-scores. Today's programmable calculators, however, can help you find and graph the actual normal curve involved in a specific problem. The program in this section computes the areas under the normal curve and the y values needed to graph the curve.



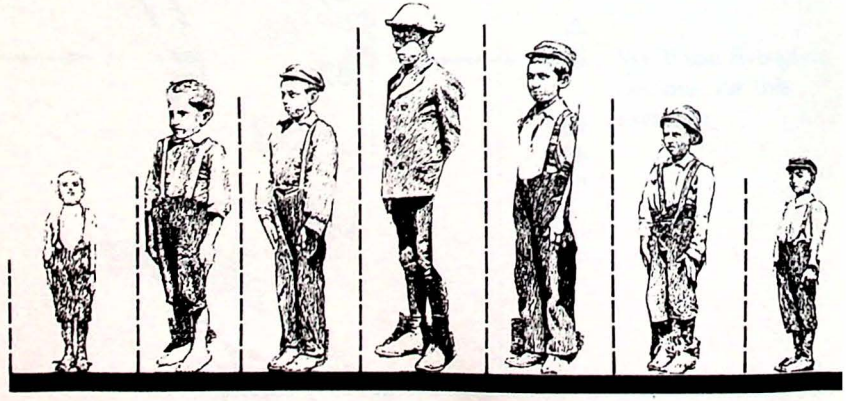
FACTS/ THEORY

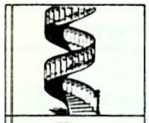
The appearance of the normal distribution for a population depends on the population mean (μ) and the population standard deviation (σ). If you know these values, you can graph the normal curve by calculating a y value for each x_i in your data set. The equation is

$$y = \left(\frac{1}{\sqrt{2\pi\sigma}}\right) e^{-(x-\mu)^2/2\sigma^2}$$

The total area under the normal curve is defined as 1. By evaluating the area under the curve between selected x_i values, we can find the probability that a sample chosen randomly from the population will fall in that range. Thus, it is very important to be able to compute these areas under the curve.

The following program solves for y and computes the areas under the normal curve using Simpson's Approximation (Master Library program ML-09).





THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl 2nd A' STO 9 − RCL 6 = x^2 +/− ÷ (2 × RCL 8 = INV ln$x$ STO 10 2 × 2nd π = $\sqrt{x}$ × RCL 7 = $1/x$ × RCL 10 = INV SBR	Calculates y (also used as f(x) subroutine in ML-09).
036	2nd Lbl 2nd E' CLR 2nd CMs R/S	Initializes program by clearing display and memories.
041	2nd Lbl A STO 6 R/S	Stores mean (μ).
046	2nd Lbl B STO 7 x^2 STO 8 R/S	Stores σ and σ^2.
054	2nd Lbl C 2nd A' 2nd Prt R/S	Loops to A' to calculate y. Displays y and stops.
059	2nd Lbl D 2nd Pgm 9 A R/S	Enters lower limit for ML-09.
065	2nd Lbl E 2nd Pgm 9 B 10 2nd Pgm 9 C 2nd Pgm 9 D 2nd Prt R/S	Enters upper limit and computes area, using n = 10 for the approximation.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}		R_{08}	σ^2
$R_{01} - R_{05}$	Used by ML-09	R_{09}	x_i
R_{06}	μ	R_{10}	Used
R_{07}	σ	R_{11}	

4 The Normal Curve

Using the Program

Procedure	Press	Display/Comments
Initialize the program.	2nd E′	0. Clears memories and display.
Set decimal for desired accuracy.	2nd Fix n	
Enter the mean and standard deviation.	A B	μ σ^2
To plot the curve, enter x_i values and calculate y_i values.	x_i C x_i C x_i C	y_i ° y_i ° y_i °
Continue this step until the desired number of y_i values have been computed.		
To find an area under the curve, enter the limits (x_i values) that define the area.		
lower limit upper limit	x_i D x_i E	x_i area ° (Simpson's approximation.)

°This value is printed if the PC-100A is used.

Sample Problem

If the mean weight of male students in college is 75 kilograms (approximately 165 pounds) and the standard deviation is 9 kilograms (about 20 pounds), draw a normal curve for the population. Then find the probability that a male student selected at random will weigh between 80 kilograms (174.6 pounds) and 85 kilograms (about 187.4 pounds).

Procedure	Press	Display/Comments
Enter program.		0
Initialize.	2nd [E]	0.
Set decimal, if desired.	2nd [fix] 4	0.0000 We'll use 4-place decimal for this example.
Enter the mean.	75 [A]	75.0000 μ
Enter the standard deviation.	9 [B]	81.0000 σ^2
Calculate y values for plotting graph. Begin by entering the mean to find the highest point of the curve.	75 [C] 80 [C] 85 [C] 90 [C] 95 [C] 100 [C] 105 [C] 110 [C]	0.0443 All other y 0.0380 values will be 0.0239 less. 0.0111 0.0038 0.0009 0.0002 0.0000

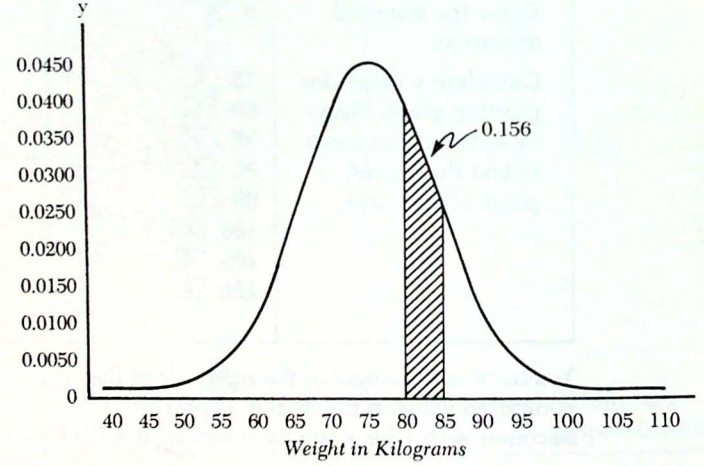
You have now computed the right side of the curve. Note that, although the computed value in the display appears to be 0, the curve actually never becomes zero. Check this by removing the fixed decimal setting.

	INV 2nd [fix]	.0000230489

Set decimal to four places again and calculate the left side of the curve. If you use values at the same distances from the mean on each side, you can expect that the curve will be symmetrical.

Procedure	Press	Display/Comments
	2nd [fix] 4	0.0000
	70 [C]	0.0380
	65 [C]	0.0239
	60 [C]	0.0111
	55 [C]	0.0038
	50 [C]	0.0009
	45 [C]	0.0002
	40 [C]	0.0000

Procedure	Press	Display/Comments
Now calculate the probability of randomly selecting a male student weighing between 80 and 85 kilograms.	80 D 85 E	80.0000 lower limit. 0.1560 Probability (the area under the curve).



Your chance of randomly selecting a male student weighing between 80 and 85 kilograms is 15.6%. As you can see from the graph, most of the area under the curve is contained between $\mu \pm 3\sigma$. This area is over 99.7% of the total area. In other words, the probability of randomly choosing a male student who weighs between 48 and 102 kilograms (about 106 and 225 pounds) is 99.7% or better.

FURTHER EXPLORATIONS

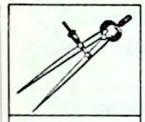
The probability of selecting a random sample that falls within the area $\mu \pm \sigma$ is given as 68%. Verify this probability with the program.

REFERENCES/ INFORMATION

M. Dean LaMont *et al, Calculator Decision-Making Sourcebook* (Dallas: Texas Instruments Incorporated, 1977), chapter 6.

Victor E. McGee, *Principles of Statistics: Traditional and Bayesian* (New York: Appleton-Century-Crofts, 1971), pp. 78-84.

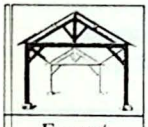
The Standard Normal Curve



OBJECTIVE

Every normal curve is symmetrical around its mean; that is, the area defined by $\mu - \sigma$ is equal to the area $\mu + \sigma$. This fact allows any normal distribution to be represented as a standard normal curve. The method is to use z-scores (also called z-values and standard scores) to transform raw sample data into a standard form. These z-scores are very important in statistical inference, and tables of z-scores are usually included in statistics textbooks. They are particularly useful when you want to compare observations obtained by different procedures.

Your programmable calculator can help you compute z-scores for your data, and Master Library program ML-14 can be used to calculate the areas under the curve, once the z-score is known.



FACTS/ THEORY

Generally, $z = \frac{x - \bar{x}}{s}$ where $\bar{x}$ is the mean of the sample and s is the standard deviation. The handy thing about z-scores is that they have a mean of 0 and a unit standard deviation. Many statistical formulas are based on standard scores instead of raw scores or deviation scores because of the ease with which z-scores can be manipulated.

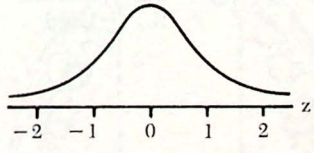
Using standard scores is effectively the same as using the standard deviation as a unit of measurement. Notice:

$$\text{If } x_i = \bar{x},\ z = \frac{\bar{x} - \bar{x}}{s} = 0$$

$$\text{If } x_i = (\bar{x} + s),\ z = \frac{(\bar{x} + s) - \bar{x}}{s} = 1$$

$$\text{If } x_i = (\bar{x} - s),\ z = \frac{(\bar{x} - s) - \bar{x}}{s} = -1$$

The standard normal curve expressed in z-scores is:



You can readily determine how a particular z-score relates to the whole curve. Since the total area under the curve is 1, the area to the right of the z-score can be interpreted as the percentage of scores on the curve greater than your particular z-score, while the area on the left can be thought of as the percentage of scores less than your z-score.

4 The Standard Normal Curve

The following program computes z-scores when x, $\bar{x}$, and s are known and also finds the areas under the curve to the right or to the left of your z-score.

The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enter learn mode.
000	[2nd] [Lbl] [2nd] [E'] [CLR] [2nd] [CMs] [R/S]	Initializes for program by clearing display and memories.
005	[2nd] [Lbl] [A] [STO] 5 [R/S] [STO] 6 [R/S] [STO] 7 [R/S]	Stores $\bar{x}$, s, and x_i (in that order).
016	[2nd] [Lbl] [B] [RCL] 7 [−] [RCL] 5 [=] [÷] [RCL] 6 [=] [STO] 9 [2nd] [Prt] [R/S]	Calculates z-score for sample element x_i.
032	[2nd] [Lbl] [C] [RCL] 9 [2nd] [Pgm] 14 [A] [2nd] [Pgm] 14 [B] [STO] 10 [2nd] [Prt] [R/S]	Computes area to the right of z.
046	[2nd] [Lbl] [D] 1 [−] [RCL] 10 [=] [2nd] [Prt] [R/S]	Computes area to the left o z.
	[LRN] [RST]	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

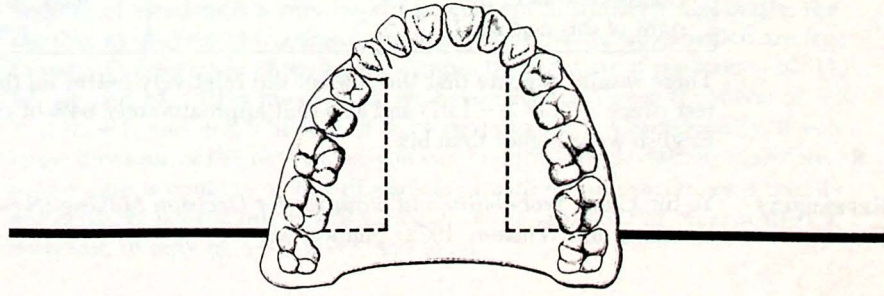
Register Contents

R_{00}		R_{06}	s
R_{01}	ML-14	R_{07}	x_i
R_{02}	ML-14	R_{08}	
R_{03}	ML-14	R_{09}	z-score
R_{04}		R_{10}	Used
R_{05}	$\bar{x}$		

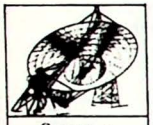
USING THE PROGRAM

Procedure	Press	Display/Comments
Initialize program.	2nd [E']	0.
Set decimal for desired accuracy.	2nd [fix] n	
Enter mean.	A	$\bar{x}$
Enter standard deviation.	R/S	s
Enter sample element (x_i).	R/S	x_i
Calculate z-score.	B	z° for x_i
Compute area to the right of z.	C	Area (z)°
Compute area to the left of z.	D	[1 − Area (z)]°

°This value is printed if the PC-100A is used.



4 The Standard Normal Curve



Sample Problem

A student scored 45 on a history test for which the mean grade was 53 and the standard deviation was 7.5. The student also scored 73 on an English test for which the mean grade was 78 with a standard deviation of 5.1. On which test was the student's relative standing higher? What percentage of students did better on the English exam?

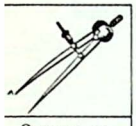
Procedure	Press	Display/Comments
Enter program.		**0**
Initialize.	2nd E′	**0.**
Set decimal, if desired.	2nd fix 2	**0.00** We'll use 2-place decimal for this example.
Enter class mean and standard deviation for the history test.	53 A 7.5 R/S	**53.00** **7.50**
Enter student's grade on test.	45 R/S	**45.00**
Calculate z-score.	B	**−1.07** z-score for history test.
Initialize.	2nd E′	**0.00**
Now enter data for the English test and calculate z-score.	78 A 5.1 R/S 73 R/S B	**78.00** $\bar{x}$ **5.10** s **73.00** x_i **−0.98** z-score for English test.
Compute area to the right of this z-score.	C	**0.84** Area to right of z-score.

These results indicate that the student did relatively better on the English test (since $-0.98 > -1.07$) and also that approximately 84% of the scores in English were higher than his.

References/ Information

Ya-lun Chou, *Probability and Statistics for Decision Making* (New York: Holt, Rinehart and Winston, 1972), chapter 13.

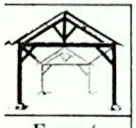
t-Distribution



OBJECTIVE

There are many times when the normal curve does not fit your needs, especially when you're working with a small sample ($n<30$). Standard scores are useful when n is large, but as n gets smaller you need another way to express a theoretical sampling distribution.

Instead of the normal distribution, a *student's t*-distribution is better for working with small samples. The following program uses the t-distribution to explore the significance of the difference between two means for independent samples.



FACTS/ THEORY

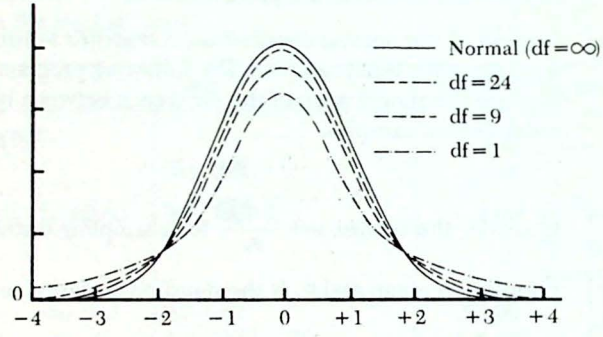
Consider the z-ratio, $z = \frac{\bar{x} - \mu}{\sigma_x}$, for a sampling distribution where μ is the population mean and σ_x is the standard deviation of the sampling distribution (usually found as $\sigma_x = \frac{\sigma}{\sqrt{n}}$, where σ is the population standard deviation). This ratio is normally distributed with a mean of 0 and unit standard deviation; however, in practice σ is usually unknown and must be estimated from the data. This estimate of σ_x is often referred to as s_x.

Let's take a look now at another useful ratio, the t-ratio:

$$t = \frac{\bar{x} - \mu}{s_x}$$

The frequency distribution of t-ratios looks like a normal curve, except that the tails are thicker as they taper off to infinity. Actually, t-ratios represent not a single distribution but rather a family of distributions which change depending upon the number of degrees of freedom (df). The number of degrees of freedom is a very important concept in statistics. Generally, the number of degrees of freedom is the number of the variables which are free to vary. For example, if we have a sample that consists of the values 10, 11, 12, 13, and 14, with a mean of 12, we have deviations from the mean of -2, -1, 0, $+1$, and $+2$. The sum of these deviations is 0. Consequently, if you know any four of the deviations, you can find the fifth deviation; therefore, df in this case is equal to 4. The df associated with sample variances is usually $n-1$, where n = the number of elements in the sample. For other statistics, however, df may be $n-2$, $n-3$, and so on.

As df gets large (approaches ∞), the t-distribution approaches the normal distribution.



A very useful test involving t-scores is the test for finding the significance of the difference between two means for independent samples. The t-value (critical t or t_c) is computed from the following formula:

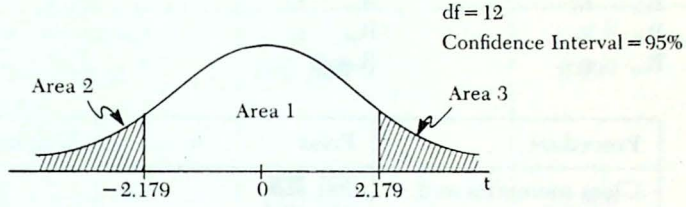
$$t_c = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{n_1 s_1^2 + n_2 s_2^2}{n_1 + n_2 - 2}\left(\frac{n_1 + n_2}{n_1 n_2}\right)}}$$

Let's say we have gathered data from two classes in different schools. We want to know whether or not the level of achievement in one school is the same as in the other, so we'll compare the mean scores for the two classes. Our testing (or null) hypothesis is

$$H_0: \bar{x}_1 = \bar{x}_2$$

If the calculated t-value falls within a 95% confidence interval on the t-curve, we'll "fail to reject" the null hypothesis. We find our confidence interval by looking up the t-value for our degrees of freedom (in this situation, $df = n_1 + n_2 - 2$) at the selected level of certainty, 95%. (See the tables at the end of this chapter.)

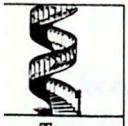
Since we are comparing two means for equality, there are two areas of rejection for H_0, and we must use a two-tailed test.



In other words, if our calculated t-value (t_c) falls within Area 1 (-2.179 to $+2.179$, for df = 12), we fail to reject H_0 and will assume that $\bar{x}_1 = \bar{x}_2$. If the t-score falls in either Area 2 or Area 3, however, we reject H_0 and assume that there is a significant difference in the levels of achievement in the two classes.

If we had been working with the hypothesis $\bar{x}_1 > \bar{x}_2$ in this example, then we would have had a directional test calling for a one-tailed test.

The program listed below allows you to enter $\bar{x}$, s, and n for two samples and then compute the critical t-value for your data.



THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 1 R/S STO 2 R/S STO 3 R/S	Stores n_1, $\bar{x}_1$, and s_1.
011	2nd Lbl B STO 4 R/S STO 5 R/S STO 6 R/S	Stores n_2, $\bar{x}_2$, and s_2.
022	2nd Lbl C RCL 1 × RCL 3 x^2 + RCL 4 × RCL 6 x^2 = + (RCL 1 + RCL 4 − 2) × ((RCL 1 + RCL 4) ÷ (RCL 1 × RCL 4)) $\sqrt{x}$ 1/x × (RCL 2 − RCL 5) = 2nd Prt R/S	Calculates critical t.
	LRN RST	Exits learn mode.

Note: Print command may be omitted if the PC-100A is not used.

4 t-Distribution

Register Contents

R_{00}		R_{04}	n_2
R_{01}	n_1	R_{05}	$\bar{x}_2$
R_{02}	$\bar{x}_1$	R_{06}	s_2
R_{03}	s_1	R_{07}	

Using the Program

Procedure	Press	Display/Comments
Clear memories and set decimal for desired accuracy.	2nd CMs 2nd Fix n	0.
Enter data.		
n_1	A	n_1
$\bar{x}_1$	R/S	$\bar{x}_1$
s_1	R/S	s_1
n_2	B	n_2
$\bar{x}_2$	R/S	$\bar{x}_2$
s_2	R/S	s_2
Calculate t_c.	C	t_c *

*This value is printed if the PC-100A is used.

Sample Problem

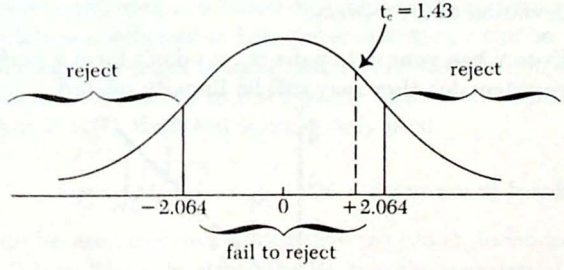
To evaluate performance on a standardized test, researchers in a school district found that a 12-member class from school A had a mean score of 87 with a standard deviation of 3.6, while a 14-member class at school B scored a mean of 85 with a standard deviation of 3.25. Is there a significant difference between the achievement levels of the two groups? (Use a 95% level of confidence.)

$H_0: \bar{x}_1 = \bar{x}_2$ The null hypothesis is that there is no difference between the groups.

$H_1: \bar{x}_1 \neq \bar{x}_2$ The alternative hypothesis is that there is a significant difference.

Procedure	Press	Display/Comments
Enter program.		0
Set decimal at 3 places.	2nd fix 3	0.000
Enter data for school A.		
n_1	12 A	12.000
$\bar{x}_1$	87 R/S	87.000
s_1	3.6 R/S	3.600
Enter data for school B.	14 B	14.000
	85 R/S	85.000
	3.25 R/S	3.250
Calculate t_c.	C	1.430

Looking up the table t-value for 24 degrees of freedom ($12+14-2=24$) and a 95% level of certainty, we find that $t=2.064$ and our confidence interval is

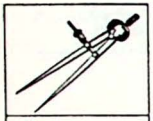


Since t_c lies in the interval -2.064 to 2.064, we cannot reject H_0. We conclude that there is no significant difference between the achievement levels of the two groups.

REFERENCES/ INFORMATION

The t-distribution and hypothesis testing are covered in most statistics textbooks. For example, see Wayne W. Daniel, *Introductory Statistics with Applications* (Boston: Houghton Mifflin Company, 1977), chapters 4 and 6.

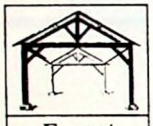
4 Linear Regression and the Correlation Coefficient



OBJECTIVE

The main task of linear regression is to find a general mathematical relationship between two sets of variables, so that the value of one variable can be used to predict the other. Linear regression, as the name implies, assumes that the two variables are related linearly; that is, if the paired data (x_i, y_i) are graphed, the points will fit (or nearly fit) a straight line. The correlation coefficient, then, reflects how well your data fit the line.

Your calculator has built-in functions to perform linear regression analysis. These functions are incorporated into the program in this section, and a t-test is added to measure the reliability of the correlation coefficient.



FACTS/ THEORY

The calculator computes a least-squares linear regression with a line described as:

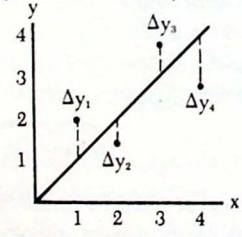
$$y = mx + b$$

where m is the slope of the line and b is the y- intercept of the line. The correlation coefficient is computed as:

$$r = \frac{m\, s_x}{s_y}$$

where s_x is the standard deviation of the x-array of data and s_y is the standard deviation of the y-array.

Even when your data pairs (x_i, y_i) don't form a perfectly straight line (and very few do), they may still be linearly related.



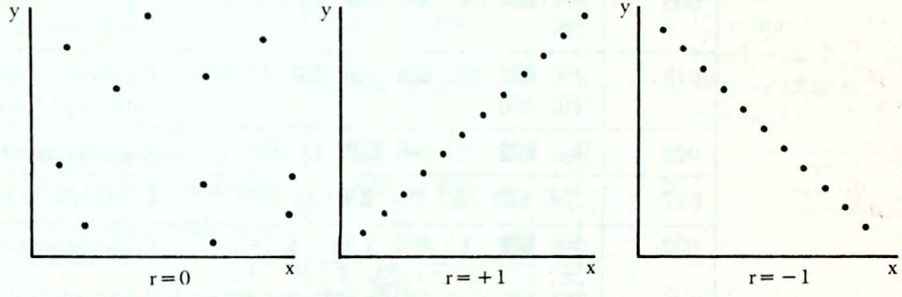
The Δy's are errors from the best-fitting straight line (the regression line) and are called residuals. Your calculator computes the slope (m) and the y-intercept (b) of the line that minimizes $\Sigma(\Delta y)^2$ for all of the (x,y) pairs entered.

$$m = \frac{\Sigma xy - \frac{\Sigma x \Sigma y}{N}}{\Sigma x^2 - \frac{(\Sigma x)^2}{N}}$$

$$b = \frac{\Sigma y - m\Sigma x}{N}$$

where N = the number of data pairs. Once the regression line is determined, you can predict the y value for any x or the x value for any y.

The correlation coefficient is expressed as a value ranging from -1 to $+1$. A negative correlation coefficient indicates a negative relationship (as x increases, y decreases, or vice versa), and a positive coefficient indicates a positive relationship. The closer the value to -1 or $+1$, the stronger the relationship is. A correlation coefficient near 0 suggests that no relationship or, at best, a random relationship exists.



The correlation coefficient, r, is based on sample data and is an estimate of the true population coefficient ρ. Like other statistics, r can be affected by sampling variations but tends to approach normal distribution as sample size increases. This fact allows you to test r by the same methods used for other statistics. When $N \leq 30$, the t-test is commonly used:

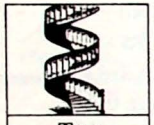
$$t = r\sqrt{\frac{N-2}{1-r^2}} \quad \text{with } N-2 \text{ degrees of freedom}$$

The t-ratio can be used to prove or disprove (within the confidence limits you select) the null hypothesis, $H_o: \rho = 0$ (that is, that the population correlation coefficient is equal to 0).

4

Linear Regression and the Correlation Coefficient

The program listed below computes the regression line for your data and calculates r, m, b, and the t-ratio.

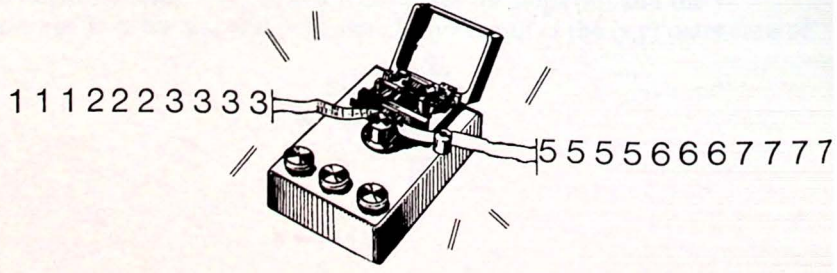


The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A x⇄t R/S 2nd Σ+ R/S GTO A	Enters x and y data.
008	2nd Lbl B 2nd Op 13 STO 7 R/S	Calculates correlation coefficient, r.
015	2nd Lbl 2nd B′ 2nd Op 12 R/S x⇄t R/S	*Optional:* calculates y-intercept (b) and slope (m).
022	2nd Lbl C 2nd Op 14 R/S	Calculates y for x value.
027	2nd Lbl D 2nd Op 15 R/S	Computes x for y value.
032	2nd Lbl E RCL 3 − 2 = ÷ (1 − RCL 7 x^2) = $\sqrt{x}$ × RCL 7 = STO 8 R/S	Calculates t-ratio.
	LRN RST	Exits learn mode.

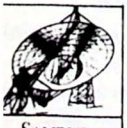
Register Contents

R_{00}			R_{07}	r
R_{01}-R_{06}	Used for summation of x, y data		R_{08}	t



USING THE PROGRAM

Procedure	Press	Display/Comments
Clear memories and set decimal for desired accuracy.	2nd CMs 2nd Fix n	0.
Enter x, y data.	x_1 A y_1 R/S	1. The calculator counts your paired-data entries.
	x_2 R/S y_2 R/S	2. (Disregard number in
	x_3 R/S y_3 R/S	3. display after x_i
	. . .	. . . entries.)
Continue until all data pairs are entered.	x_n R/S y_n R/S	N
Calculate correlation coefficient, r.	B	r
If desired, calculate y-intercept and slope of the line.	2nd B' R/S	b m
To calculate y for x value, enter x.	C	y
To calculate x for y value, enter y.	D	x
Calculate t-score.	E	t



SAMPLE PROBLEM

1. The QZ Company, a manufacturer of small parts requiring hand assembly, is looking for a way to predict how well prospective employees will work out on its assembly line. The personnel director of the company finds a widely used test for measuring manual dexterity and hand-to-eye coordination and administers the test to 10 new assembly-line employees. Then he checks on their daily unit output at the end of three months and collects the following data:

4

Linear Regression and the Correlation Coefficient

EMPLOYEE	*TEST SCORE* (*x*)	*DAILY OUTPUT* (*in 100 units*) (*y*)
James	8.0	3.4
Doris	7.5	2.9
Julia	9.0	3.5
Robert	5.5	2.5
Amy	6.0	2.7
Max	4.0	2.6
Carl	9.5	3.6
Jane	4.5	2.6
Louis	8.5	3.1
Richard	5.0	3.0

Does there seem to be a correlation between scores on the test and daily output at the end of three months? If so, what should the test cut-off score be if the company wants its employees to be producing at least 260 units a day at the end of three months? In general, what daily output can the company expect (at the end of three months) from employees who score 8.5 on the test?

Procedure	Press	Display/Comments
Enter program.		**0**
Set decimal for desired accuracy.	[2nd] [fix] 2	**0.00** We'll use 2-place decimal.
Enter (x,y) data.		
James	8 [A] 3.4 [R/S]	**1.00** Note that, after
Doris	7.5 [R/S] 2.9 [R/S]	**2.00** subroutine [A]
Julia	9 [R/S] 3.5 [R/S]	**3.00** has been
Robert	5.5 [R/S] 2.5 [R/S]	**4.00** initialized for first
Amy	6 [R/S] 2.7 [R/S]	**5.00** x entry, all other
Max	4 [R/S] 2.6 [R/S]	**6.00** data is entered by
Carl	9.5 [R/S] 3.6 [R/S]	**7.00** pressing [R/S].
Jane	4.5 [R/S] 2.6 [R/S]	**8.00**
Louis	8.5 [R/S] 3.1 [R/S]	**9.00**
Richard	5 [R/S] 3 [R/S]	**10.00**
Calculate r.	[B]	**0.87** Correlation coefficient.

Procedure	Press	Display/Comments
Calculate score that would indicate output of 260 units daily (260 ÷ 100 = 2.6).	2.6 D	4.52 Score (predicted x value for y = 2.6).
Calculate predicted output for persons scoring 8.5 on test.	8.5 C	3.30 Output (y where x = 8.5).

Since the correlation coefficient is 0.87, the personnel director feels there is a pretty good relationship between the test score and the unit output at the end of 3 months. The regression analysis also indicates that prospective employees should score at least 4.5 on the test if the daily output at the end of 3 months is expected to be 260 units, and that those who score 8.5 on the test will generally be able to achieve an output of about 330 units a day at the end of the training period.

Realizing that he has a pretty small sample here, the director plans to continue his testing program and compile further data. If the relationship holds, he should be able to develop some efficient hiring guidelines.

2. For students taking a basic statistics course, the instructor has compiled the following data:

Student	*IQ*	*Final Exam Grade*
1	120	87
2	131	91
3	112	85
4	135	93
5	111	73
6	118	82
7	122	85
8	125	89
9	108	74
10	127	88

The instructor wants to know if there is a correlation between students' IQ's and final exam scores in his class. To answer his question, find r and then perform the t-test using a 95% level of certainty.

Procedure	Press	Display/Comments
Clear memories and display.	CLR 2nd CMS	0.00
Enter data.	120 A 87 R/S	1.00
	131 R/S 91 R/S	2.00
	112 R/S 85 R/S	3.00
	135 R/S 93 R/S	4.00
	111 R/S 73 R/S	5.00
	118 R/S 82 R/S	6.00
	122 R/S 85 R/S	7.00
	125 R/S 89 R/S	8.00
	108 R/S 74 R/S	9.00
	127 R/S 88 R/S	10.00
Calculate r.	B	0.89 Correlation coefficient.
Calculate t-score.	E	5.55 t-score.

There appears to be a good positive relationship. Now that you have your t-score, you want to test two possibilities:

$$H_0: \rho = 0$$
$$H_1: \rho \neq 0$$

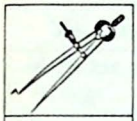
Thus, this is a two-tailed test. By consulting a table of t-scores (see the tables at the end of this chapter), you find that the range of t-scores for a 95% level of certainty with 8 degrees of freedom is ± 2.306. Therefore, if your t-score falls within this range (-2.306 to $+2.306$), you accept the null hypothesis — that is, you are 95% sure that your correlation coefficient of 0.89 has been affected by sampling variations, and there is really not much correlation between IQ and final exam grades. Since your t-score is not in this range, however, you reject the null hypothesis and are 95% sure that r approximates ρ correctly.

REFERENCES/ INFORMATION

Wayne W. Daniel, *Introductory Statistics with Applications* (Boston: Houghton Mifflin Company, 1977), chapter 9.

Robert D. Mason, *Statistical Techniques in Business and Economics*, 3rd ed. (Homewood, Illinois: Richard D. Irwin, Inc., 1974), chapter 18.

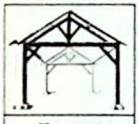
Analysis of Variance



OBJECTIVE

One of the most useful and powerful tools in statistics is the analysis of variance (ANOVA). Generally ANOVA is a model which divides observed variation of data into different parts. You can then assess the magnitude of the variation and use an F-test to help you decide if the variation is greater than expected under the null hypothesis. The ANOVA model can be used to compare two or more processes, concepts, products, results, or means.

The following program will generate an F-value for a one-way analysis of variance.



FACTS/ THEORY

There are two variations which need to be considered, the variances *between* the groups (MS_b) and the variances *within* the groups (MS_w). The observed F-value (F_o) is found as follows:

$$F_o = \frac{MS_b}{MS_w}$$

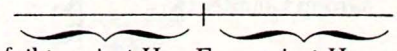
$$MS_b = \frac{\frac{(\Sigma x_1)^2}{N_1} + \frac{(\Sigma x_2)^2}{N_2} + \ldots + \frac{(\Sigma x_n)^2}{N_n} - \frac{(\Sigma x_t)^2}{N_t}}{K-1}$$

$$MS_w = \frac{\Sigma x_1^2 + \Sigma x_2^2 + \ldots \Sigma x_n^2 - \left[\frac{(\Sigma x_1)^2}{N_1} + \frac{(\Sigma x_2)^2}{N_2} + \ldots + \frac{(\Sigma x_n)^2}{N_n}\right]}{N_t - K}$$

Where N_t = the total number of observations,
N_i = the number of observations in each group, and
K = the number of groups.

Once you've found F_o you need to compare this value with a critical F-value (F_c) found in a table of F-values (see the F-table at the end of this chapter). The number of degrees of freedom (df) for MS_b (the numerator) is K-1, and df for MS_w (the denominator) is $N_t - K$. To use the table, find df for the numerator, and then come down the column until you reach df for the denominator.

The following comparison can then be made:



If F_o is to the left of F_c, you have failed to reject H_o. If F_o is to the right of F_c, you reject H_o.

Procedure	Press	Display/Comments
Clear memories and display.	CLR 2nd CMs	0.00
Enter data.	120 A 87 R/S	1.00
	131 R/S 91 R/S	2.00
	112 R/S 85 R/S	3.00
	135 R/S 93 R/S	4.00
	111 R/S 73 R/S	5.00
	118 R/S 82 R/S	6.00
	122 R/S 85 R/S	7.00
	125 R/S 89 R/S	8.00
	108 R/S 74 R/S	9.00
	127 R/S 88 R/S	10.00
Calculate r.	B	0.89 Correlation coefficient.
Calculate t-score.	E	5.55 t-score.

There appears to be a good positive relationship. Now that you have your t-score, you want to test two possibilities:

$$H_0: \rho = 0$$
$$H_1: \rho \neq 0$$

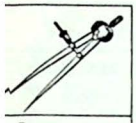
Thus, this is a two-tailed test. By consulting a table of t-scores (see the tables at the end of this chapter), you find that the range of t-scores for a 95% level of certainty with 8 degrees of freedom is ± 2.306. Therefore, if your t-score falls within this range (-2.306 to $+2.306$), you accept the null hypothesis – that is, you are 95% sure that your correlation coefficient of 0.89 has been affected by sampling variations, and there is really not much correlation between IQ and final exam grades. Since your t-score is not in this range, however, you reject the null hypothesis and are 95% sure that r approximates ρ correctly.

REFERENCES/ INFORMATION

Wayne W. Daniel, *Introductory Statistics with Applications* (Boston: Houghton Mifflin Company, 1977), chapter 9.

Robert D. Mason, *Statistical Techniques in Business and Economics*, 3rd ed. (Homewood, Illinois: Richard D. Irwin, Inc., 1974), chapter 18.

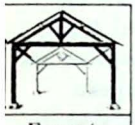
Analysis of Variance



OBJECTIVE

One of the most useful and powerful tools in statistics is the analysis of variance (ANOVA). Generally ANOVA is a model which divides observed variation of data into different parts. You can then assess the magnitude of the variation and use an F-test to help you decide if the variation is greater than expected under the null hypothesis. The ANOVA model can be used to compare two or more processes, concepts, products, results, or means.

The following program will generate an F-value for a one-way analysis of variance.



FACTS/ THEORY

There are two variations which need to be considered, the variances *between* the groups (MS_b) and the variances *within* the groups (MS_w). The observed F-value (F_o) is found as follows:

$$F_o = \frac{MS_b}{MS_w}$$

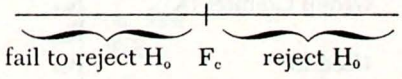
$$MS_b = \frac{\frac{(\Sigma x_1)^2}{N_1} + \frac{(\Sigma x_2)^2}{N_2} + \ldots + \frac{(\Sigma x_n)^2}{N_n} - \frac{(\Sigma x_t)^2}{N_t}}{K-1}$$

$$MS_w = \frac{\Sigma x_1^2 + \Sigma x_2^2 + \ldots \Sigma x_n^2 - \left[\frac{(\Sigma x_1)^2}{N_1} + \frac{(\Sigma x_2)^2}{N_2} + \ldots + \frac{(\Sigma x_n)^2}{N_n}\right]}{N_t - K}$$

Where N_t = the total number of observations,
N_i = the number of observations in each group, and
K = the number of groups.

Once you've found F_o you need to compare this value with a critical F-value (F_c) found in a table of F-values (see the F-table at the end of this chapter). The number of degrees of freedom (df) for MS_b (the numerator) is K-1, and df for MS_w (the denominator) is $N_t - K$. To use the table, find df for the numerator, and then come down the column until you reach df for the denominator.

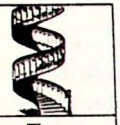
The following comparison can then be made:



If F_o is to the left of F_c, you have failed to reject H_o. If F_o is to the right of F_c, you reject H_o.

4

Analysis of Variance



THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 7 2nd Op 20 R/S	Stores number of observations in group (N_i) and counts groups.
007	2nd Lbl 2nd A' 2nd Σ+ R/S 2nd Dsz 7 2nd A' RCL 1 SUM 10 x^2 ÷ RCL 3 SUM 12 = SUM 13 RCL 2 SUM 11 CLR 2nd Pgm 1 SBR CLR R/S	Enters x_i data for group and checks to see if all x_i data have been entered. If not, returns to A' to enter next x_i. If all x_i's have been entered for group, goes on to calculate Σx_i, $\frac{(\Sigma x_i)^2}{N_i}$, ΣN_i, and Σx_i^2, and then clears R_i through R_6 for next group's entry.
037	2nd Lbl B RCL 13 − RCL 10 x^2 ÷ RCL 12 = ÷ (RCL 0 − 1) = STO 14 2nd Prt R/S	Calculates MS_b.
061	2nd Lbl C RCL 11 − RCL 13 = ÷ (RCL 12 − RCL 0) = STO 15 2nd Prt R/S	Calculates MS_w.
082	2nd Lbl D RCL 14 ÷ RCL 15 = 2nd Prt R/S	Calculates F_o.
092	2nd Lbl E CLR 2nd CMs R/S	Initializes by clearing display and memories.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}	Group Counter (K)	R_{07}	N_i	R_{14}	MS_b
R_{01}		R_{08}		R_{15}	MS_w
R_{02}	Used for	R_{09}		R_{16}	
R_{03}	statistics	R_{10}	Σx_i	R_{17}	
R_{04}	entries	R_{11}	Σx_i^2	R_{18}	
R_{05}		R_{12}	ΣN_i	R_{19}	
R_{06}		R_{13}	$\frac{\Sigma(\Sigma x_i)^2}{N_i}$	R_{20}	

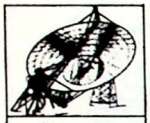
USING THE PROGRAM

Procedure	Press	Display/Comments
Initialize program.	E	0. Clears memories and display.
Set decimal for desired accuracy.	2nd fix n	
Enter data for first group.		
Number of observations	N_1 A	N_1
x_1	x_1 R/S	1. Data entries
x_2	x_2 R/S	2. counted.
• • •	• • •	• • •
x_n	x_n R/S	n
When all observations for this group have been entered, be sure to press R/S so that future computations can be made.	R/S	0. Computes and sums data for calculation of MS_b and MS_w.
Enter data for next group.		
N_2	N_2 A	N_2
x_1	x_1 R/S	x_1
x_2	x_2 R/S	x_2
• • •	• • •	• • •
x_n	x_n R/S	x_n
Compile data from second group.	R/S	0.
Repeat these steps for each group of data.		
N_n	N_n A	N_n
x_1	x_1 R/S	x_1
x_2	x_2 R/S	x_2
• • •	• • •	• • •
x_n	x_n R/S	x_n
	R/S	0.
Compute MS_b.	B	MS_b*
Compute MS_w.	C	MS_w*
Compute F_o.	D	F_o*

*This value is printed if the PC-100A is used.

4

Analysis of Variance



Sample Problem

A food company was testing four package designs for a new product. Fifteen stores were selected, and each store was randomly assigned one of the package designs. Other relevant conditions, such as location and number of packages on the shelf, price, and promotional efforts, were kept the same for all the stores. The sales during the study period were:

Package Designs

1	*2*	*3*	*4*
13	15	20	22
17	11	18	30
15	14	22	19
	16	17	
	18		

Is there a difference in the average sales for the four package designs?

The null hypothesis is

H_0: There is no difference in average sales for the four designs.

Our number of degrees of freedom for MS_b is 3 ($K-1=4-1=3$), and df for MS_w is 11 ($N_t-K=15-4=11$). Looking in the F-table, we find the F-value for our degrees of freedom is 3.59 at a 95% level of confidence. Now, we calculate F_o.

Procedure	Press	Display/Comments
Enter program.		**0**
Set two-place decimal.	2nd fix 2	**0.00**
Enter data for first design.		
$N_1=3$	3 A	**3.00**
$x_1=13$	13 R/S	**1.00**
$x_2=17$	17 R/S	**2.00**
$x_3=15$	15 R/S	**3.00**
Compile data for calculation of MS_b and MS_w.	R/S	**0.00**

Procedure	Press	Display/Comments
Enter data for second design.		
$N_2 = 5$	5 [A]	5.00
$x_1 = 15$	15 [R/S]	1.00
$x_2 = 11$	11 [R/S]	2.00
$x_3 = 14$	14 [R/S]	3.00
$x_4 = 16$	16 [R/S]	4.00
$x_5 = 18$	18 [R/S]	5.00
Compile data.	[R/S]	0.00
Enter data for third design.		
$N_3 = 4$	4 [A]	4.00
$x_1 = 20$	20 [R/S]	1.00
$x_2 = 18$	18 [R/S]	2.00
$x_3 = 22$	22 [R/S]	3.00
$x_4 = 17$	17 [R/S]	4.00
Compile data.	[R/S]	0.00
Enter data for fourth design.		
$N_4 = 3$	3 [A]	3.00
$x_1 = 22$	22 [R/S]	1.00
$x_2 = 30$	30 [R/S]	2.00
$x_3 = 19$	19 [R/S]	3.00
Compile data.	[R/S]	0.00
Compute MS_b.	[B]	60.06
Compute MS_w.	[C]	10.38
Compute F_o.	[D]	5.78

Now we can make our decision:

fail to reject H_0	reject H_0
	$F_o = 5.78$
$F_c = 3.59$	

Since $5.78 > 3.59$, we must reject H_0 and assume that there is a significant difference in average sales.

REFERENCES/ INFORMATION

Robert D. Mason, *Statistical Techniques in Business and Economics*, 3rd ed. (Homewood, Illinois: Richard D. Irwin, Inc., 1974), pp. 362-366.

4 Table A

t values for one-tailed test (For checking only an upper or a lower limit)

Degrees of Freedom	Level of Certainty 90%	95%	99%	99.5%
1	3.078	6.314	31.821	63.657
2	1.886	2.920	6.965	9.925
3	1.638	2.353	4.541	5.841
4	1.533	2.132	3.747	4.604
5	1.476	2.015	3.365	4.032
6	1.440	1.943	3.143	3.707
7	1.415	1.895	2.998	3.499
8	1.397	1.860	2.896	3.355
9	1.383	1.833	2.821	3.250
10	1.372	1.812	2.764	3.169
11	1.363	1.796	2.718	3.106
12	1.356	1.782	2.681	3.055
13	1.350	1.771	2.650	3.012
14	1.345	1.761	2.624	2.977
15	1.341	1.753	2.602	2.947
16	1.337	1.746	2.583	2.921
17	1.333	1.740	2.567	2.898
18	1.330	1.734	2.552	2.878
19	1.328	1.729	2.539	2.861
20	1.325	1.725	2.528	2.845
21	1.323	1.721	2.518	2.831
22	1.321	1.717	2.508	2.819
23	1.319	1.714	2.500	2.807
24	1.318	1.711	2.492	2.797
25	1.316	1.708	2.485	2.787
26	1.315	1.706	2.479	2.779
27	1.314	1.703	2.473	2.771
28	1.313	1.701	2.467	2.763
29	1.311	1.699	2.462	2.756
30	1.310	1.697	2.457	2.750
40	1.303	1.684	2.423	2.704
60	1.296	1.671	2.390	2.660
120	1.289	1.658	2.358	2.617
∞	1.282	1.645	2.326	2.576

Table B

t values for two-tailed test (For checking both upper and lower limits)

Degrees of Freedom	Level of Certainty				
	80%	90%	95%	99%	99.9%
1	3.078	6.314	12.706	63.657	636.619
2	1.886	2.920	4.303	9.925	31.598
3	1.638	2.353	3.182	5.841	12.941
4	1.533	2.132	2.776	4.604	8.610
5	1.476	2.015	2.571	4.032	6.859
6	1.440	1.943	2.447	3.707	5.959
7	1.415	1.895	2.365	3.499	5.405
8	1.397	1.860	2.306	3.355	5.041
9	1.383	1.833	2.262	3.250	4.781
10	1.372	1.812	2.228	3.169	4.587
11	1.363	1.796	2.201	3.106	4.437
12	1.356	1.782	2.179	3.055	4.318
13	1.350	1.771	2.160	3.012	4.221
14	1.345	1.761	2.145	2.977	4.140
15	1.341	1.753	2.131	2.947	4.073
16	1.337	1.746	2.120	2.921	4.015
17	1.333	1.740	2.110	2.898	3.965
18	1.330	1.734	2.101	2.878	3.922
19	1.328	1.729	2.093	2.861	3.883
20	1.325	1.725	2.086	2.845	3.850
21	1.323	1.721	2.080	2.831	3.819
22	1.321	1.717	2.074	2.819	3.792
23	1.319	1.714	2.069	2.807	3.767
24	1.318	1.711	2.064	2.797	3.745
25	1.316	1.708	2.060	2.787	3.725
26	1.315	1.706	2.056	2.779	3.707
27	1.314	1.703	2.052	2.771	3.690
28	1.313	1.701	2.048	2.763	3.674
29	1.311	1.699	2.045	2.756	3.659
30	1.310	1.697	2.042	2.750	3.646
40	1.303	1.684	2.021	2.704	3.551
60	1.296	1.671	2.000	2.660	3.460
120	1.289	1.658	1.980	2.617	3.373
∞	1.282	1.645	1.960	2.576	3.291

Table C

Values of F for 95% Level of Certainty

Degrees of Freedom of the Numerator

Degrees of Freedom of the Denominator	1	2	3	4	5	6	7	8	9	10	12	15	20	30	60	120	∞
1	161.4	199.5	215.7	224.6	230.2	234.0	236.8	238.9	240.5	241.9	243.9	245.9	248.0	250.1	252.2	253.3	254.3
2	18.51	19.00	19.16	19.25	19.30	19.33	19.35	19.37	19.38	19.40	19.41	19.43	19.45	19.46	19.48	19.49	19.50
3	10.13	9.55	9.28	9.12	9.01	8.94	8.89	8.85	8.81	8.79	8.74	8.70	8.66	8.62	8.57	8.55	8.53
4	7.71	6.94	6.59	6.39	6.26	6.16	6.09	6.04	6.00	5.96	5.91	5.86	5.80	5.75	5.69	5.66	5.63
5	6.61	5.79	5.41	5.19	5.05	4.95	4.88	4.82	4.77	4.74	4.68	4.62	4.56	4.50	4.43	4.40	4.36
6	5.99	5.14	4.76	4.53	4.39	4.28	4.21	4.15	4.10	4.06	4.00	3.94	3.87	3.81	3.74	3.70	3.67
7	5.59	4.74	4.35	4.12	3.97	3.87	3.79	3.73	3.68	3.64	3.57	3.51	3.44	3.38	3.30	3.27	3.23
8	5.32	4.46	4.07	3.84	3.69	3.58	3.50	3.44	3.39	3.35	3.28	3.22	3.15	3.08	3.01	2.97	2.93
9	5.12	4.26	3.86	3.63	3.48	3.37	3.29	3.23	3.18	3.14	3.07	3.01	2.94	2.86	2.79	2.75	2.71
10	4.96	4.10	3.71	3.48	3.33	3.22	3.14	3.07	3.02	2.98	2.91	2.85	2.77	2.70	2.62	2.58	2.54
11	4.84	3.98	3.59	3.36	3.20	3.09	3.01	2.95	2.90	2.85	2.79	2.72	2.65	2.57	2.49	2.45	2.40
12	4.75	3.89	3.49	3.26	3.11	3.00	2.91	2.85	2.80	2.75	2.69	2.62	2.54	2.47	2.38	2.34	2.30
13	4.67	3.81	3.41	3.18	3.03	2.92	2.83	2.77	2.71	2.67	2.60	2.53	2.46	2.38	2.30	2.25	2.21
14	4.60	3.74	3.34	3.11	2.96	2.85	2.76	2.70	2.65	2.60	2.53	2.46	2.39	2.31	2.22	2.18	2.13
15	4.54	3.68	3.29	3.06	2.90	2.79	2.71	2.64	2.59	2.54	2.48	2.40	2.33	2.25	2.16	2.11	2.07
16	4.49	3.63	3.24	3.01	2.85	2.74	2.66	2.59	2.54	2.49	2.42	2.35	2.28	2.19	2.11	2.06	2.01
17	4.45	3.59	3.20	2.96	2.81	2.70	2.61	2.55	2.49	2.45	2.38	2.31	2.23	2.15	2.06	2.01	1.96
18	4.41	3.55	3.16	2.93	2.77	2.66	2.58	2.51	2.46	2.41	2.34	2.27	2.19	2.11	2.02	1.97	1.92
19	4.38	3.52	3.13	2.90	2.74	2.63	2.54	2.48	2.42	2.38	2.31	2.23	2.16	2.07	1.98	1.93	1.88
20	4.35	3.49	3.10	2.87	2.71	2.60	2.51	2.45	2.39	2.35	2.28	2.20	2.12	2.04	1.95	1.90	1.84
21	4.32	3.47	3.07	2.84	2.68	2.57	2.49	2.42	2.37	2.32	2.25	2.18	2.10	2.01	1.92	1.87	1.81
22	4.30	3.44	3.05	2.82	2.66	2.55	2.46	2.40	2.34	2.30	2.23	2.15	2.07	1.98	1.89	1.84	1.78
23	4.28	3.42	3.03	2.80	2.64	2.53	2.44	2.37	2.32	2.27	2.20	2.13	2.05	1.96	1.86	1.81	1.76
24	4.26	3.40	3.01	2.78	2.62	2.51	2.42	2.36	2.30	2.25	2.18	2.11	2.03	1.94	1.84	1.79	1.73
25	4.24	3.39	2.99	2.76	2.60	2.49	2.40	2.34	2.28	2.24	2.16	2.09	2.01	1.92	1.82	1.77	1.71
26	4.23	3.37	2.98	2.74	2.59	2.47	2.39	2.32	2.27	2.22	2.15	2.07	1.99	1.90	1.80	1.75	1.69
27	4.21	3.35	2.96	2.73	2.57	2.46	2.37	2.31	2.25	2.20	2.13	2.06	1.97	1.88	1.79	1.73	1.67
28	4.20	3.34	2.95	2.71	2.56	2.45	2.36	2.29	2.24	2.19	2.12	2.04	1.96	1.87	1.77	1.71	1.65
29	4.18	3.33	2.93	2.70	2.55	2.43	2.35	2.28	2.22	2.18	2.10	2.03	1.94	1.85	1.75	1.70	1.64
30	4.17	3.32	2.92	2.69	2.53	2.42	2.33	2.27	2.21	2.16	2.09	2.01	1.93	1.84	1.74	1.68	1.62
40	4.08	3.23	2.84	2.61	2.45	2.34	2.25	2.18	2.12	2.08	2.00	1.92	1.84	1.74	1.64	1.58	1.51
60	4.00	3.15	2.76	2.53	2.37	2.25	2.17	2.10	2.04	1.99	1.92	1.84	1.75	1.65	1.53	1.47	1.39
120	3.92	3.07	2.68	2.45	2.29	2.17	2.09	2.02	1.96	1.91	1.83	1.75	1.66	1.55	1.43	1.35	1.25
∞	3.84	3.00	2.60	2.37	2.21	2.10	2.01	1.94	1.88	1.83	1.75	1.67	1.57	1.46	1.32	1.22	1.00

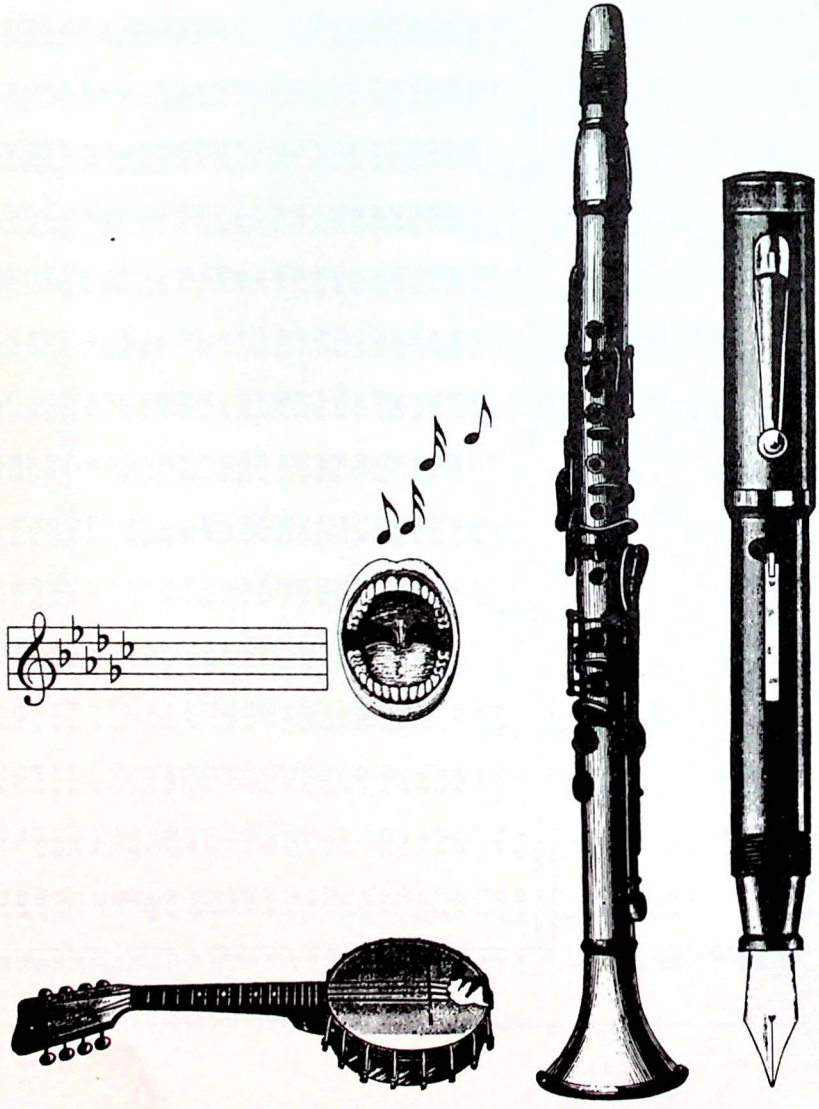
Table D

Values of F for 99% Level of Certainty

Degrees of Freedom of the Numerator

Degrees of Freedom of the Denominator	1	2	3	4	5	6	7	8	9	10	12	15	20	30	60	120	∞
1	4052	4999.5	5403	5625	5764	5859	5928	5982	6022	6056	6106	6157	6209	6261	6313	6339	6366
2	98.50	99.00	99.17	99.25	99.30	99.33	99.36	99.37	99.39	99.40	99.42	99.43	99.45	99.47	99.48	99.49	99.50
3	34.12	30.82	29.46	28.71	28.24	27.91	27.67	27.49	27.35	27.23	27.05	26.87	26.69	26.50	26.32	26.22	26.13
4	21.20	18.00	16.69	15.98	15.52	15.21	14.98	14.80	14.66	14.55	14.37	14.20	14.02	13.84	13.65	13.56	13.46
5	16.26	13.27	12.06	11.39	10.97	10.67	10.46	10.29	10.16	10.05	9.89	9.72	9.55	9.38	9.20	9.11	9.02
6	13.75	10.92	9.78	9.15	8.75	8.47	8.26	8.10	7.98	7.87	7.72	7.56	7.40	7.23	7.06	6.97	6.88
7	12.25	9.55	8.45	7.85	7.46	7.19	6.99	6.84	6.72	6.62	6.47	6.31	6.16	5.99	5.82	5.74	5.65
8	11.26	8.65	7.59	7.01	6.63	6.37	6.18	6.03	5.91	5.81	5.67	5.52	5.36	5.20	5.03	4.95	4.86
9	10.56	8.02	6.99	6.42	6.06	5.80	5.61	5.47	5.35	5.26	5.11	4.96	4.81	4.65	4.48	4.40	4.31
10	10.04	7.56	6.55	5.99	5.64	5.39	5.20	5.06	4.94	4.85	4.71	4.56	4.41	4.25	4.08	4.00	3.91
11	9.65	7.21	6.22	5.67	5.32	5.07	4.89	4.74	4.63	4.54	4.40	4.25	4.10	3.94	3.78	3.69	3.60
12	9.33	6.93	5.95	5.41	5.06	4.82	4.64	4.50	4.39	4.30	4.16	4.01	3.86	3.70	3.54	3.45	3.36
13	9.07	6.70	5.74	5.21	4.86	4.62	4.44	4.30	4.19	4.10	3.96	3.82	3.66	3.51	3.34	3.25	3.17
14	8.86	6.51	5.56	5.04	4.69	4.46	4.28	4.14	4.03	3.94	3.80	3.66	3.51	3.35	3.18	3.09	3.00
15	8.68	6.36	5.42	4.89	4.56	4.32	4.14	4.00	3.89	3.80	3.67	3.52	3.37	3.21	3.05	2.96	2.87
16	8.53	6.23	5.29	4.77	4.44	4.20	4.03	3.89	3.78	3.69	3.55	3.41	3.26	3.10	2.93	2.84	2.75
17	8.40	6.11	5.18	4.67	4.34	4.10	3.93	3.79	3.68	3.59	3.46	3.31	3.16	3.00	2.83	2.75	2.65
18	8.29	6.01	5.09	4.58	4.25	4.01	3.84	3.71	3.60	3.51	3.37	3.23	3.08	2.92	2.75	2.66	2.57
19	8.18	5.93	5.01	4.50	4.17	3.94	3.77	3.63	3.52	3.43	3.30	3.15	3.00	2.84	2.67	2.58	2.49
20	8.10	5.85	4.94	4.43	4.10	3.87	3.70	3.56	3.46	3.37	3.23	3.09	2.94	2.78	2.61	2.52	2.42
21	8.02	5.78	4.87	4.37	4.04	3.81	3.64	3.51	3.40	3.31	3.17	3.03	2.88	2.72	2.55	2.46	2.36
22	7.95	5.72	4.82	4.31	3.99	3.76	3.59	3.45	3.35	3.26	3.12	2.98	2.83	2.67	2.50	2.40	2.31
23	7.88	5.66	4.76	4.26	3.94	3.71	3.54	3.41	3.30	3.21	3.07	2.93	2.78	2.62	2.45	2.35	2.26
24	7.82	5.61	4.72	4.22	3.90	3.67	3.50	3.36	3.26	3.17	3.03	2.89	2.74	2.58	2.40	2.31	2.21
25	7.77	5.57	4.68	4.18	3.85	3.63	3.46	3.32	3.22	3.13	2.99	2.85	2.70	2.54	2.36	2.27	2.17
26	7.72	5.53	4.64	4.14	3.82	3.59	3.42	3.29	3.18	3.09	2.96	2.81	2.66	2.50	2.33	2.23	2.13
27	7.68	5.49	4.60	4.11	3.78	3.56	3.39	3.26	3.15	3.06	2.93	2.78	2.63	2.47	2.29	2.20	2.10
28	7.64	5.45	4.57	4.07	3.75	3.53	3.36	3.23	3.12	3.03	2.90	2.75	2.60	2.44	2.26	2.17	2.06
29	7.60	5.42	4.54	4.04	3.73	3.50	3.33	3.20	3.09	3.00	2.87	2.73	2.57	2.41	2.23	2.14	2.03
30	7.56	5.39	4.51	4.02	3.70	3.47	3.30	3.17	3.07	2.98	2.84	2.70	2.55	2.39	2.21	2.11	2.01
40	7.31	5.18	4.31	3.83	3.51	3.29	3.12	2.99	2.89	2.80	2.66	2.52	2.37	2.20	2.02	1.92	1.80
60	7.08	4.98	4.13	3.65	3.34	3.12	2.95	2.82	2.72	2.63	2.50	2.35	2.20	2.03	1.84	1.73	1.60
120	6.85	4.79	3.95	3.48	3.17	2.96	2.79	2.66	2.56	2.47	2.34	2.19	2.03	1.86	1.66	1.53	1.38
∞	6.63	4.61	3.78	3.32	3.02	2.80	2.64	2.51	2.41	2.32	2.18	2.04	1.88	1.70	1.47	1.32	1.00

5 Music Theory



Introduction

Ancient Greek and Chinese philosophers theorized about the relationship of mathematics to music. According to Plato:

> It was music that brought order and control into human experiences in the evolution of the race and in this way prepared for the functioning of the rational faculty.

However, musicians had to wait until the middle of the twentieth century for computer technology to develop and assist them with the mathematical aspects of music.

About 1948, digital computers and other electronic data-processing machines were used for research in musicology, such as Bertrand Bronson's folksong research, and in music education. Experiments in musical composition with the aid of a computer began in 1955-56 with the ILLIAC automatic high-speed digital computer at the University of Illinois. In 1957 Lejaren Hiller and Leonard Isaacson used the ILLIAC computer to compose *Illiac Suite for String Quartet*, the first "computer composition."

David H. Cope in *New Directions in Music* divided composer use of the computer into five categories:

1. computer-composed music in terms of "human-imposed parameters" (computer selection from programmed styles);
2. computer-generated sound wherein the composer probably has the most control over his materials;
3. computer aid for random or probability theory constructs;
4. direct control of synthesizer functions by the computer;
5. visual (CRT) notation for direct analog output.[1]

Computer composition involved programming the basic material and stylistic parameters for the composition. *Illiac Suite for String Quartet* was composed in this way, and even though such compositions have not been artistically successful, they have attracted attention to the possibilities of the computer.

In 1959 Max Matthews and James Tenney from Bell Telephone Laboratories collaborated with J. K. Randall and Hubert Howe from Princeton to produce the first computer program for generation of sound, MUSIC IVB. This technique allowed the composer almost complete control over the composition.

[1]David H. Cope, *New Directions in Music*, 2nd ed. (Dubuque, Iowa: Wm. C. Brown Company Publishers, 1976), p. 100.

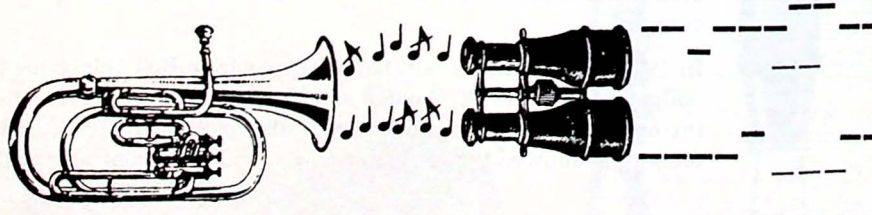
5

Introduction

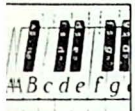
Computer aid in compositions based on random or probability theory constructs was used by Iannis Xenakis to produce ST/10-1, 080262 in 1962. The computer did not actually compose this music, but only aided the composer with high-speed computations; however, research in the field of computer control of synthesizer functions has produced several machines that can be used by the composer. The Synthi 100's Digital Sequencer 256 (E.M.S.) can store 10,240 items capable of controlling six different parameters simultaneously over a sequence of 256 events. The *Sal-Mar Construction* made by Salvatore Martirano is semiportable and capable of producing live performances of computer-synthesizer music.

Developments in computer technology have a continuing effect on computer music. Because of the computer's high-speed capacity and accuracy, it has extended the physical limits of composer and performer alike. Music that would have been extremely tedious for the composer to create can now be accomplished in a relatively short time. The range of performance has been greatly extended by the control and accuracy of the computer. Errors in performance are virtually nonexistent in some areas of computer music. Also, sounds that were unavailable with traditional instruments can be produced by pushing a button.

The extended programming, memory, and printing capabilities of programmable calculators have brought many aspects of computer music into the range of the hand-held calculator. The remainder of this chapter will show how your calculator can be used in music theory for learning music intervals, analysis of music, and composition.



Intervals

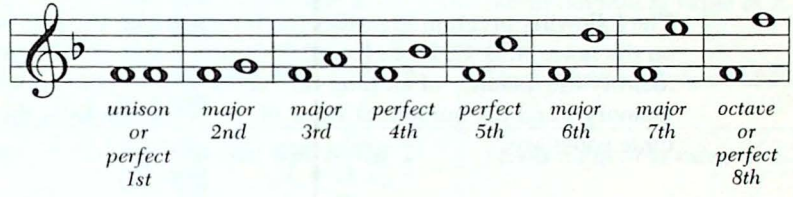


OBJECTIVE

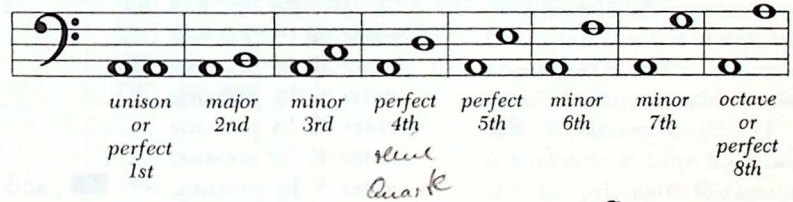
An interval is the distance in pitch between two notes. In analyzing intervals, two items need to be determined: quality (major, minor, augmented, diminished, and perfect) and quantity (first, second, third, fourth, fifth, sixth, seventh, etc.). The program included here analyzes the intervals between tonic and the other notes of the scale and identifies the quantity (from first through seventh) of the interval.

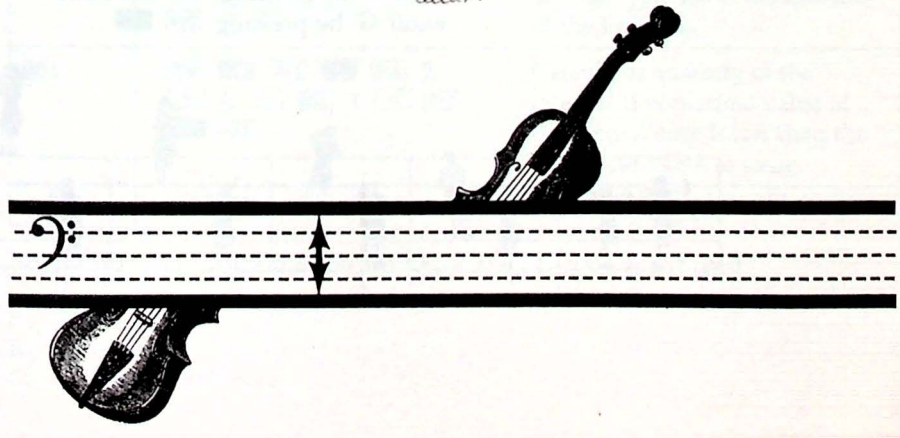
FACTS/ THEORY

In major keys the intervals between tonic and the other degrees of the scale have the following qualities and quantities:



Quantities in minor keys are the same as in the parallel major key, but the qualities are sometimes different.





5

INTERVALS

Although this program does not involve the qualities of intervals, you should keep in mind that quality plays an important part in the analysis of intervals. In general the patterns for identifying quality are:

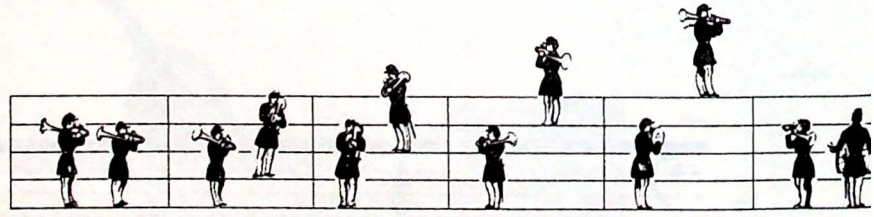
1. a major interval decreased by a semitone is minor.
2. a minor interval decreased by a semitone is diminished.
3. a major interval increased by a semitone is augmented.
4. a perfect interval decreased by a semitone is diminished.
5. a perfect interval increased by a semitone is augmented.

The following program identifies the intervals one through seven, operating on the notes A, B, C, D, E, F, and G. It uses a system of numerical codes to identify the quantity of an interval, based on the tonic code stored in memory 1 and the numerical value of the second note in the interval. The tonic codes are:

C = 1	G = 5
D = 2	A = 6
E = 3	B = 7
F = 4	

For convenience in entering the second note in the interval, each note of the scale is identified with a user-defined label, so that you

enter A by pressing [A],
enter B by pressing [B],
enter C by pressing [C],
enter D by pressing [D],
enter E by pressing [E],
enter F by pressing [2nd] [D'], and
enter G by pressing [2nd] [E'].



THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl C 1 STO 2 2nd B′	Stores numerical value of C.
006	2nd Lbl D 2 STO 2 2nd B′	Stores numerical value of D.
012	2nd Lbl E 3 STO 2 2nd B′	Stores numerical value of E.
018	2nd Lbl 2nd B′ 4 STO 2 2nd B′	Stores numerical value of F.
024	2nd Lbl 2nd E′ 5 STO 2 2nd B′	Stores numerical value of G.
030	2nd Lbl A 6 STO 2 2nd B′	Stores numerical value of A.
036	2nd Lbl B 7 STO 02	Stores numerical value of B.
041	2nd Lbl 2nd B′ RCL 1 x≷t RCL 2 INV 2nd x=t 2nd A′ RCL 2 − RCL 1 + 1 = 2nd Prt R/S	Tests second note against tonic. If numerical value of second note is less than the numerical value of tonic, transfers to 2nd A′. If numerical value of second note is more than value of tonic, calculates the quantity of the interval.
061	2nd Lbl 2nd A′ RCL 2 + 8 − RCL 1 = 2nd Prt R/S	Calculates quantity of the interval if numerical value of the second note is less than the numerical value of tonic.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}
R_{01} tonic value
R_{02} value of second note

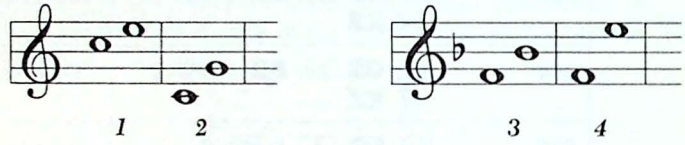
5 Intervals

Using the Program

Procedure	Press	Display/Comments
Enter tonic code in memory 1.	code STO 01	code
Press user-defined label key assigned to second note.	A or B or C or D or E or 2nd D' or 2nd E'	quantity°

°This value is printed if the PC-100A is used.

Sample Problem

Identify the following intervals:

Procedure	Press	Display/Comments
Enter program.		0
1. C is tonic, so enter 1 in memory 1.	1 STO 01	1. Tonic code.
Enter second note, E.	E	3. The interval is a third.
2. Since the value of C is already stored, enter only the second note, G.	2nd E'	5. Interval is a fifth.
3. Enter numerical code for tonic, F.	4 STO 01	4. Tonic code.
Enter second note, B.	B	4. Interval is a fourth.
4. Enter second note, E.	E	7. Interval is a seventh.

FURTHER EXPLORATIONS

1. Rewrite the program to calculate lower intervals; that is, intervals leading down from a note, rather than upward.
2. Write a program to identify the quality, as well as quantity, of intervals.
3. Rewrite the program to calculate inversions; that is, intervals that are inverted from top to bottom.

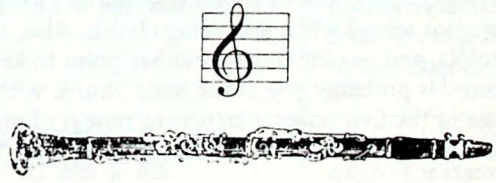
REFERENCES/ INFORMATION

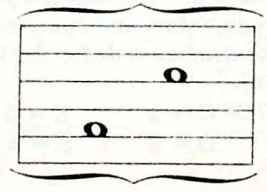
For additional information about intervals, see:

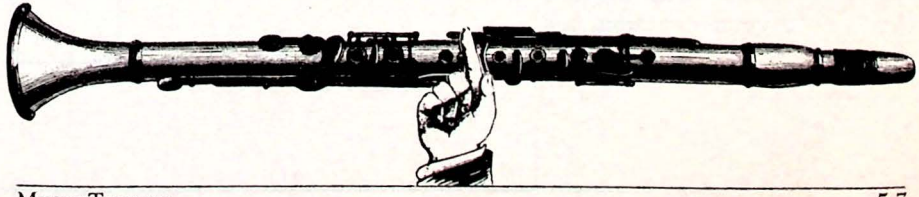
William Christ, Richard DeLone, Vernon Kliewer, Lewis Rowell, and William Thomson, *Materials and Structure of Music I* (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1966).

Tom Manoff, *The Music Kit* (New York: W. W. Norton, 1976).

Allen Winold and John Rehm, *Introduction to Music Theory* (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1971).







5 Chord Identification

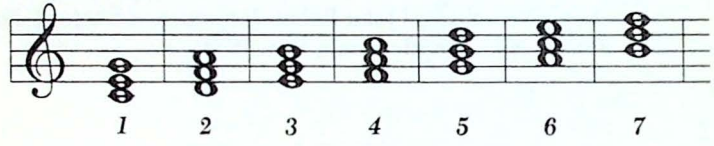
OBJECTIVE

A chord consists of three or more notes sounding at the same time, and this combination of tones is called harmony. In a piece of music each chord has a specific function, determined by the key in which the music is written. Identifying chords plays an important role in harmonic analysis and in transposing music from one key to another.

The program in this section analyzes a three-note chord (triad) and identifies it as a one, two, three, four, five, six, or seven chord. (These are often indicated with Roman numerals I, II, III, IV, V, VI, and VII.) Note that the program automatically transposes the notes of the chord to root position before totaling the scale degrees of the notes to find the chord number.

FACTS/ THEORY

In the key of C, chords built on each scale degree are:



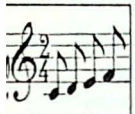
Like intervals, chords are also labeled as major, minor, augmented, or diminished. The rules for identifying these qualities may vary from one music theory department to another. Be sure to keep in mind the rules established by your school when analyzing chords. Also, be alert for suspension, pedal points, and passing tones. Another point to keep in mind is that a four-note chord is probably one of the basic chords with a seventh added. (Note that if one of the four notes is an octave repeat of one of the other notes in the chord, it is not a four-note chord.)

As in the section on Intervals, the program for identifying chords uses a numerical code system and operates on the notes A, B, C, D, E, F, and G. You'll first enter the number code for the key in which the music is written, and these codes are:

C = 1	E = 3	G = 5	B = 7
D = 2	F = 4	A = 6	

Then enter the three notes of the chord by pressing the user-defined label key assigned to each note. That is, you enter

A by pressing [A],
B by pressing [B],
C by pressing [C],
D by pressing [D],
E by pressing [E],
F by pressing [2nd] [D'], and/or
G by pressing [2nd] [E'].

Once your data is entered, you are ready to calculate the number value of the chord, one through seven. *Note: The program will not correctly identify chords other than these seven.*

THE PROGRAM

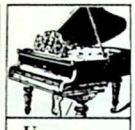
Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [2nd] [A'] [2nd] [CMs] [STO] 1 [x⇄t] [CLR] [R/S]	Clears memories and stores key code in memory 1 and t register.
008	[2nd] [Lbl] [A] 6 [2nd] [x≥t] [EE]	Enters A and tests against key code. If value of A ≥ key value, transfers to subroutine [EE].
013	[2nd] [Lbl] [2nd] [C'] [+] 8 [−] [RCL] 1 [=] [SUM] 2 [R/S]	If less, continues to [2nd] [C'], where A is transposed and summed to memory 2.
024	[2nd] [Lbl] [B] 7	Enters value of B (since 7 is largest note value, it does not have to be tested against key value). Then goes on to transpose note value and sums to memory 2.
027	[2nd] [Lbl] [EE] [−] [RCL] 1 [+] 1 [=] [SUM] 2 [R/S]	

Chord Identification

Step #	Program Keystrokes	Purpose/Comments
038	2nd Lbl C 1 2nd x≥t EE 2nd C'	Enters and tests value of C, transferring to proper subroutine for transposition and summation.
044	2nd Lbl D 2 2nd x≥t EE 2nd C'	Enters and tests value of D.
050	2nd Lbl E 3 2nd x≥t EE 2nd C'	Enters and tests value of E.
056	2nd Lbl 2nd D' 4 2nd x≥t EE 2nd C'	Enters and tests value of F.
062	2nd Lbl 2nd E' 5 2nd x≥t EE 2nd C'	Enters and tests value of G.
068	2nd Lbl 2nd B' RCL 2 x:t 9 2nd x≥t x^2 10 2nd x≥t y^x 11 2nd x≥t $\sqrt{x}$ 12 2nd x≥t $1/x$ 13 2nd x≥t ln x 14 2nd x≥t CE 3 SBR CLR	Tests chord value for identification. Note that x^2, y^x, $\sqrt{x}$, ln x, CE, and CLR are used here as subroutine labels. Identifies chord as a three if all tests default.
099	2nd Lbl x^2 1 SBR CLR	A one chord.
104	2nd Lbl y^x 6 SBR CLR	A six chord.
109	2nd Lbl $\sqrt{x}$ 4 SBR CLR	A four chord.
114	2nd Lbl $1/x$ 2 SBR CLR	A two chord.
119	2nd Lbl ln x 7 SBR CLR	A seven chord.
124	2nd Lbl CE 5	A five chord.
127	2nd Lbl CLR 2nd Prt R/S RCL 1 2nd CMs STO 1 x:t CLR R/S	Displays chord value and stops. Then clears memories and restores key code in memory 1 for next chord evaluation.
	LRN RST	Exits learn mode.

Note: Print command may be omitted if the PC-100A is not used.

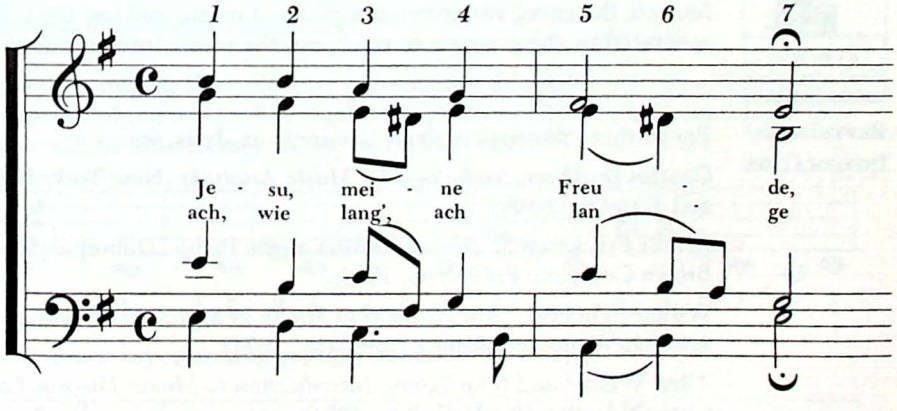
REGISTER CONTENTS

R_{00}
R_{01} key code
R_{02} chord value
R_{03}
R_{04}

USING THE PROGRAM

Procedure	Press	Display/Comments
Enter key code.	code 2nd A'	0
Enter each note in chord, one at a time.	A or B or C or D or E or 2nd D' or 2nd E'	The transposed value of the note will appear in the display after the appropriate label key is pushed.
Calculate chord value.	2nd B'	chord value°
Prepare for next chord.	R/S	0

°This value is printed if the PC-100A is used.

SAMPLE PROBLEM

Analyze the marked chords in this excerpt from Bach's *Jesu, Meine Freude.* The key is E minor.

Chord Identification

Procedure	Press	Display/Comments
Enter program.		0
Enter key code, 3.	3 2nd A'	0
Enter notes of first chord.	E 2nd E' B	1. 3. 5.
Calculate chord value.	2nd B'	1. A one chord.
Prepare for next chord.	R/S	0
Enter notes of second chord.	D E 2nd D'	2.
Calculate chord value.	2nd B'	5. A five chord.
Prepare for next chord.	R/S	0

Continue in this way until all the chords have been analyzed.

The chords will be:

Chord 1 = one	Chord 4 = one	Chord 7 = one
Chord 2 = five	Chord 5 = two	
Chord 3 = four	Chord 6 = five	

Further Explorations

Analyze the chord structures of a piece of music and use the information generated by the program to transpose the music into another key.

References/ Information

For further information about harmonic analysis, see:

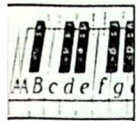
Charles Burkhart, *Anthology for Music Analysis* (New York: Holt, Rinehart and Winston, 1965).

Arthur Frackenpohl, *Harmonization at the Piano* (Dubuque, Iowa: Wm. C. Brown Company Publishers, 1969).

William Thomson, *Introduction to Music as Structure* (Reading, Mass.: Addison-Wesley Publishing Company, 1971).

Allen Winold and John Rehm, *Introduction to Music Theory* (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1971).

Sight-Singing and Melodic Contours



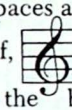
OBJECTIVE

Sight-singing is a sophisticated and valuable skill developed by extensive practice, and melodic contours are a form of music analysis useful in determining how a melody will sound by looking at the notes on the staff.

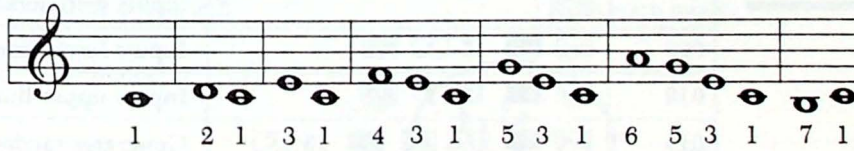
The following program will generate random notes for practice in sight-singing and, if the PC-100A is used, will print a melodic contour of the notes generated.

FACTS/ THEORY

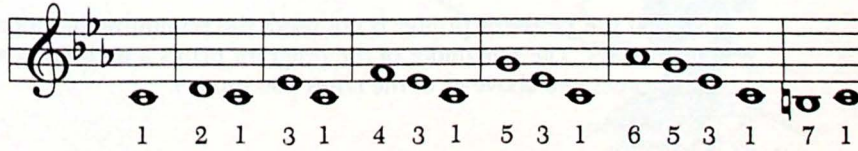
The "fixed *do*" system assigns *do* to 1 or tonic of each key. ("Moveable *do*" assigns *do* to the note C. For example, in the key of C major (which we use in this section) C = 1, D = 2, E = 3, F = 4, G = 5, A = 6, and B = 7.

The music staff is made of five horizontal parallel lines, and the pitch of any note is determined by its location on the lines or spaces between the lines of the staff. The names of the various lines and spaces are assigned by the clef sign on the left side of the staff. The treble clef,  , or G-clef assigns the note G above middle C to the second line from the bottom of the staff. All other pitches are relative to the G. Base clef, , or F-clef assigns the note F below middle C to the second line from the top of the staff. All other pitches are relative to the F. Alto clef, , a C-clef, assigns middle C to the middle line to the staff. Tenor clef, , another C-clef, assigns middle C to the fourth line from the bottom of the staff.

Practice singing the following patterns first by singing the intervals in each measure of the music. Eventually the pattern will be so well established that you will be able to sing the entire melody easily.



For minor keys a similar pattern should be practiced:



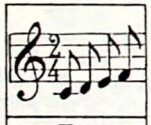
5

Sight-Singing and Melodic Contours

In the following program the first twenty-one steps utilize the Master Library random number generator (ML-15) to create a series of numbers corresponding to notes. As mentioned, we have used the scale of C major with note values of 1 = C, 2 = D, 3 = E, 4 = F, 5 = G, 6 = A, and 7 = B. The program can easily be expanded to create a larger range of notes simply by raising the upper limit (see program step 012).

The remainder of the program is designed to be used with the PC-100A to print the letter name of the note in its relative position on the tape, thus forming a melodic contour. Read it by tearing the paper from the PC-100A and turning it so that the first notes printed are on the left. By cutting the contour into sections corresponding to musical phrases, each section of the music can be analyzed separately. Again, the program can be adapted beyond the seven-note range simply by adding additional printing subroutines and rearranging the printing subroutine calls in the program.

You can use the program in three ways. First, it creates a random melody that offers practice in sight-singing. Next, the program will print the melodic contour of these randomly generated notes for analysis. And finally, you can control the printing of a melodic contour by entering the number of any note and then pressing [C]. This procedure allows you to print out a melody you create or to print a melodic contour for pre-written melodies (within the selected range of notes).



The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [A] [2nd] [Pgm] 15 [E] [R/S]	Selects program ML-15 and inputs seed number.
006	[2nd] [Pgm] 15 [A] [R/S]	Inputs lower limit.
010	[2nd] [Pgm] 15 [B] [R/S]	Inputs upper limit.
014	[2nd] [Lbl] [B] [2nd] [Pgm] 15 [C] [2nd] [Int] [R/S]	Generates random number.

(*Note:* If the PC-100A printer is not used, the remaining program steps are unnecessary. The remainder of the program prints a melodic contour of the notes generated above or of the notes you input.)

Step #	Program Keystrokes	Purpose/Comments
021	2nd Lbl C x:t 1 2nd x=t 2nd C' + 1 = 2nd x=t 2nd D' + 1 = 2nd x=t 2nd E' + 1 = 2nd x=t D + 1 = 2nd x=t E + 1 = 2nd x=t 2nd A' + 1 = 2nd x=t 2nd B' R/S	Enters generated note value in t register; then tests for value (C = 1, D = 2, E = 3, F = 4, G = 5, A = 6, and B = 7). When x = t, program goes to print routine and prints name of note in relative scale position.
058	2nd Lbl 2nd A' 2nd Op 00 13 2nd Op 3 GTO EE	Print codes for A.
068	2nd Lbl 2nd B' 2nd Op 00 14000000 2nd Op 4 GTO EE	Print codes for B.
084	2nd Lbl 2nd C' 2nd Op 00 15 2nd Op 1 GTO EE	Print codes for C.
094	2nd Lbl 2nd D' 2nd Op 00 16000000 2nd Op 2 GTO EE	Print codes for D.
110	2nd Lbl 2nd E' 2nd Op 00 1700 2nd Op 2 GTO EE	Print codes for E.
122	2nd Lbl D 2nd Op 00 2100000000 2nd Op 3 GTO EE	Print codes for F.
140	2nd Lbl E 2nd Op 00 220000 2nd Op 3 GTO EE	Print codes for G.
154	2nd Lbl EE 2nd Op 5 2nd Adv R/S	Print command subroutine.
	LRN RST	Exits learn mode.

5

Sight-Singing and Melodic Contours

Register Contents

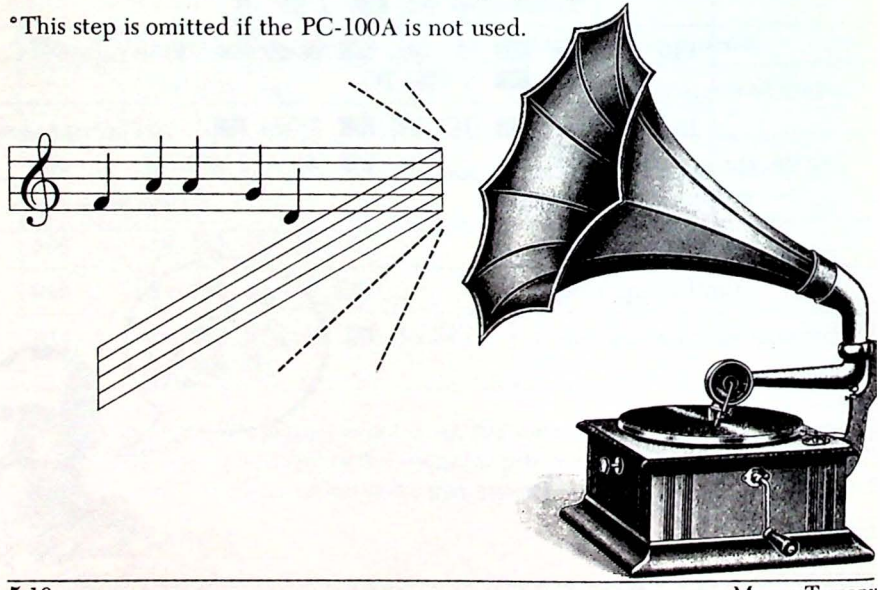
R_{00}		R_{05}	Used	R_{10}	Lower Limit
R_{01}	Used	R_{06}	Used	R_{11}	Upper Limit
R_{02}	Used	R_{07}	Used		
R_{03}	Used	R_{08}			
R_{04}	Used	R_{09}	Used		

Using the Program

Procedure	Press	Display/Comments
Data Entry: 1. Enter seed number ($0 \leq seed \leq 199017$).	seed [A]	seed
2. Enter lower limit (x). 3. Enter upper limit (x+n).	x [R/S] (x+n) [R/S]	x (x+n)
Generate random number (note value).	[B]	note value
Print note.°	[C]	Prints letter name of note in its relative position.

Continue to press [B] to generate new notes (if the PC-100A is used, press [C] after each note value appears in the display to print melodic contour).

°This step is omitted if the PC-100A is not used.



SAMPLE PROBLEM

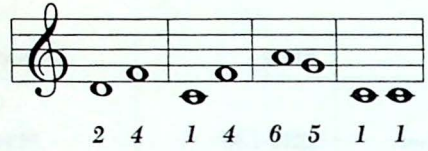
Example 1: Using a seed number of 1234, generate an 8-note melody with a range of middle C to B.

Procedure	Press	Display/Comments
Enter program.		0
Enter seed number.	1234 [A]	1234.
Enter limits:		
Lower limit (middle c) = 1.	1 [R/S]	1.
Upper limit = 1+7(b).	8 [R/S]	8.
Generate random number (note value).	[B]	2. (D)
Print note.*	[C]	16000000. (D print code)
(Press [B] to generate each new note, and press [C] to print it, if the PC-100A is used.)	[B]	4. (F)
	[C]	2100000000. (F print code)
	[B]	1. (C)
	[C]	15. (C print code)
	[B]	4.
	[C]	2100000000.
	[B]	6. (A)
	[C]	13. (A print code)
	[B]	5. (G)
	[C]	220000. (G print code)
	[B]	1. (C)
	[C]	15.
	[B]	1. (C)
	[C]	15.

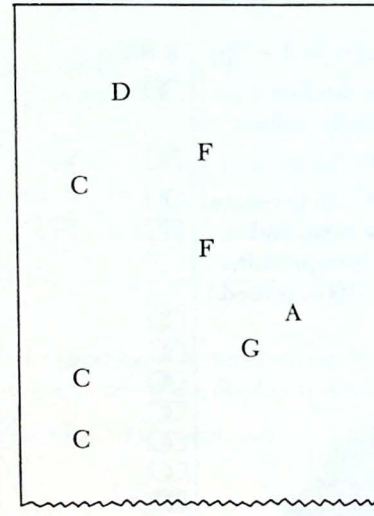
*Omit this step if the PC-100A is not used.

Sight-Singing and Melodic Contours

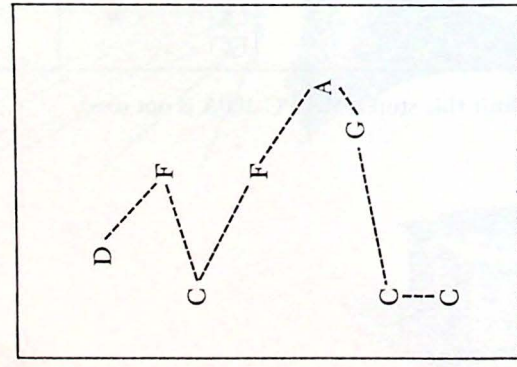
In the key of C major the melody is:



The PC-100A print-out will look like this:



Turning the tape so that the beginning note is on your left, you can analyze the melodic contour.



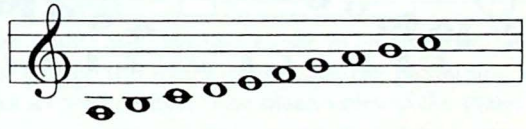
Example 2: Print the following melodic contour and identify this famous melody:

1, 1, 5, 5, 6, 6, 5, 4, 4, 3, 3, 2, 2, 1.

Procedure	Press	Display	Print-out
	1 C	15.	C
	1 C	15.	C
	5 C	220000.	G
	5 C	220000.	G
	6 C	13.	A
	6 C	13.	A
	5 C	220000.	G
	4 C	2100000000.	F
	4 C	2100000000.	F
	3 C	1700.	E
	3 C	1700.	E
	2 C	16000000.	D
	2 C	16000000.	D
	1 C	15.	C

FURTHER EXPLORATIONS

Generate a 20-note melody with a range of 10 notes (A below middle C to C above middle C) and print the melodic contour. Use the following note values:



1 2 3 4 5 6 7 8 9 10

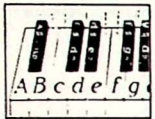
REFERENCES/ INFORMATION

Paul Hindemith, *Elementary Training for Musicians* (New York: Associated Music Publishers, Inc., 1949).

George A. Wedge, *Advanced Ear-Training and Sight-Singing* (New York: G. Schirmer, Inc., 1922).

Master Library, pp. 52–54.

5 Composition



OBJECTIVE

The program used in the section on sight-singing and melodic contours is actually a form of composition — the creation of a melody line. The first twenty-one steps of that program (the random-number-generating portion) can also be used to create entire compositions — melody and harmony — when certain harmonizing techniques are applied.

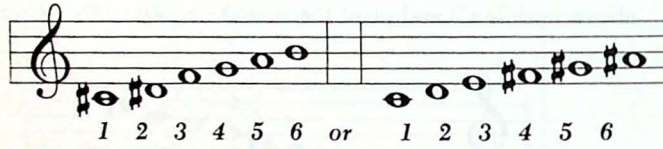
Since the program and instructions for using it are given in the section on sight-singing and melodic contours, we won't repeat these here. Instead, we'll discuss certain scales and techniques used in compositions and give examples as we go along.

Note: If you are using the PC-100A printer, you may wish to insert a print command (2nd Prt) just before R/S in the program.

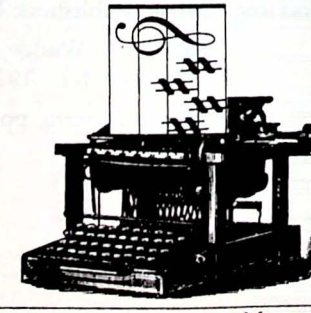
FACTS/ THEORY

WHOLE-TONE SCALE

Possibly the easiest scale to use in composition is the whole-tone scale because it has no half-tone dissonances. Therefore, the random number generator can be used to compose melodies and harmonies without fear of unpleasant sounds.



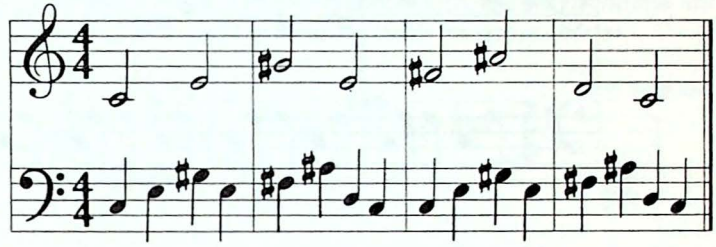
Once a series of notes on this scale has been generated by the program, you can create a composition by making each note a half note for melody and using the same series in quarter notes (diminution) repeated twice for harmony. Unfortunately, whole-tone compositions tend to be a bit boring, but the practice in composition is worthwhile.

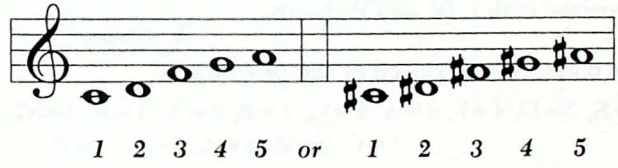


SAMPLE PROBLEM

Using a lower limit of 1 and an upper limit of 7 (1 + 6), the program produced this series of notes:

1 = C, 3 = E, 5 = G #, 3 = E, 4 = F #, 6 = A #, 2 = D, and 1 = C.

Following the directions above, this short piano composition resulted:

PENTATONIC SCALES

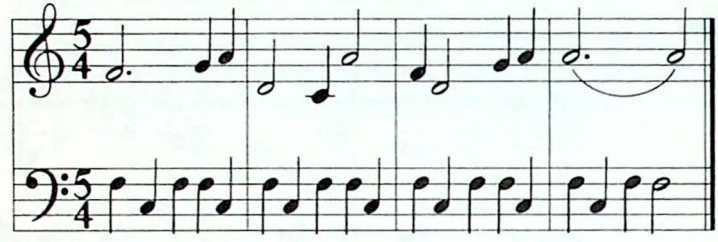
Pentatonic scales are also easy to use in composition because there are no half-steps. Unlike the whole-tone scale, the pentatonic scale rapidly establishes a tonal center. (The black notes of the piano form a pentatonic scale.)

To harmonize the pentatonic melody resulting from the program, you can use a repeating bass (or ostinato) pattern.

5 Composition

Sample Problem

With lower and upper limits of 1 and 6 (1 + 5), the program gave this series:

3 = F, 4 = G, 5 = A, 2 = D, 1 = C, 5 = A, 3 = F, 2 = D, 4 = G, 5 = A, and 5 = A.

An F - C ostinato was added for accompaniment on the piano.

Major and Minor Scales

A few very simple rules need to be followed for a conventional-sounding composition.

1. Tonic, first scale degree or 1 from the calculator, should be used on the first beat of the first complete measure.
2. To establish a strong cadence (ending), the next-to-last note should be from the V^7 chord (a five chord with a seventh interval added).
3. The last note should also be tonic. If the melody is harmonized, the last note may be any note in the one (I) chord and the lowest-sounding note of the harmony should be tonic.

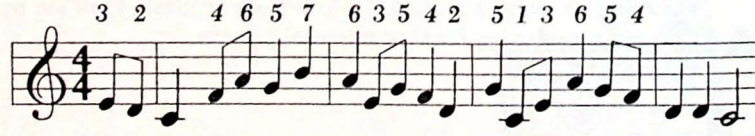
The lower and upper limits used in the program should be 1 and 8 (1 + 7) respectively, and the melody produced by the program can easily be harmonized with I, IV, and V chords.

Sample Problem

These notes were produced by the program:

3 = E, 2 = D, 4 = F, 6 = A, 5 = G, 7 = B, 6 = A, 3 = E, 5 = G, 4 = F, 2 = D, 5 = G,
1 = C, 3 = E, 6 = A, 5 = G, 4 = F.

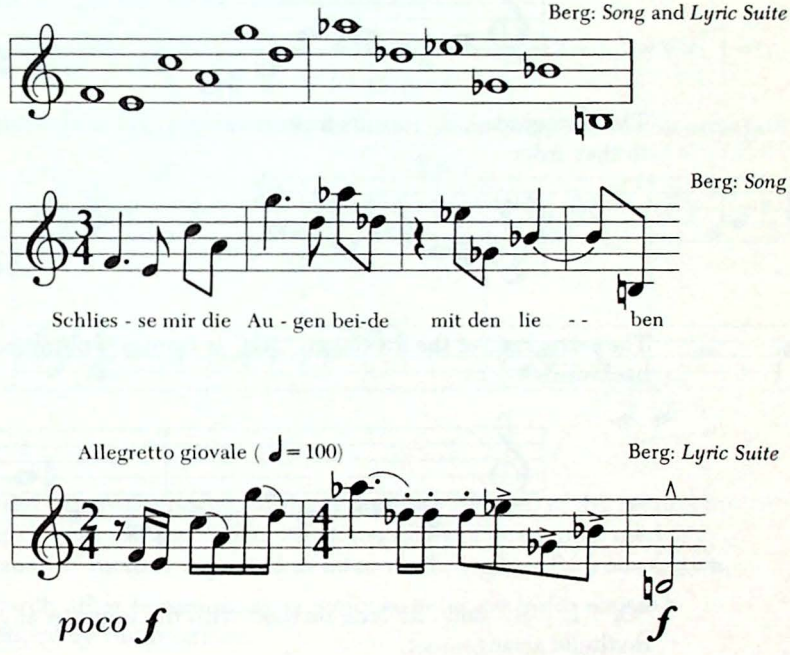
This melody resulted from applying the above rules:



Adding the appropriate I, IV, and V chords will provide harmony for your melody.

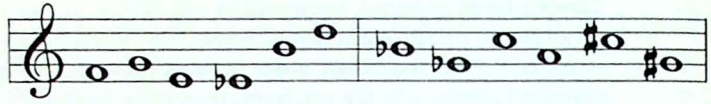
Twelve-Tone Row

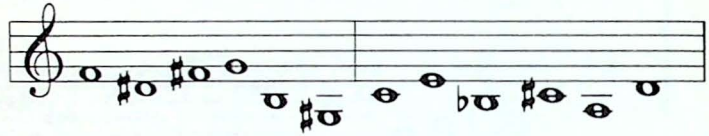
Twelve-tone, or serial, composition can also be accomplished by using the program to produce a series of notes. The twelve-tone row uses all twelve notes in the octave in any order without repeating notes. (When you generate notes with the program, remember to discard repeated notes as they appear.)

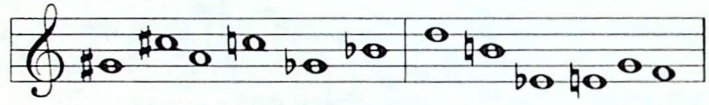
Berg's first row was used in two compositions, the song *Schliesse mir die Augen beide* (1925) and *Lyric Suite* (1926) for string quartet.

Berg: *Song* and *Lyric Suite*
Berg: *Song*
Schlies - se mir die Au - gen bei-de mit den lie -- ben
Allegretto giovale (♩ = 100)
Berg: *Lyric Suite*
poco f
f

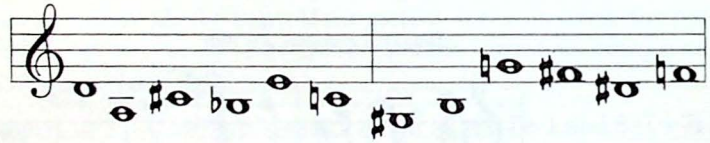
Berg and other twelve-tone composers of that time constructed their tone rows with great care to achieve specific purposes. A calculator-created row, however, can also offer some interesting possibilities. And, once the original row has been constructed, three other forms can be derived from it: inversion, retrograde, and retrograde of inversion.

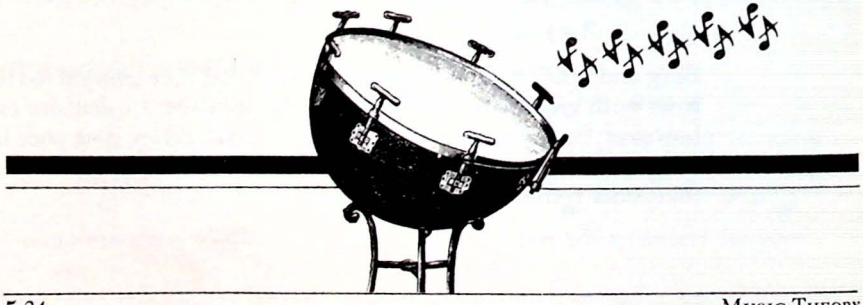
For example, consider this original row, "O," from Webern's Opus 23:



The inversion, "I," is obtained by inverting the direction, up to down and down to up, of each interval from the first note of the original:

The retrograde, "R," results from reversing "O" and writing the notes down in that order:



The retrograde of the inversion, "RI," is similarly obtained by writing "I" backwards:

"O," "I," "R," and "RI" can be used with the notes in any octave and in any rhythmic arrangement.



SAMPLE PROBLEM

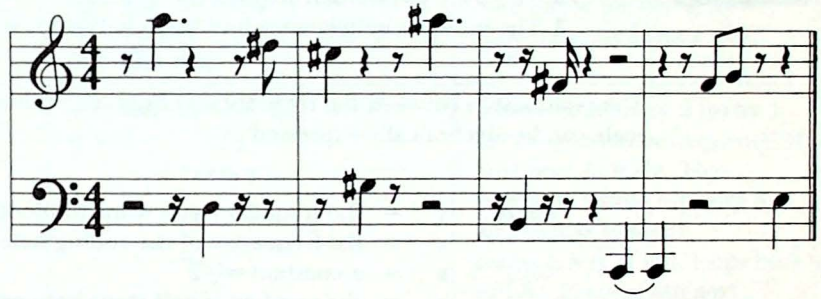
For a twelve-tone row, use these note values and limits for the random number program:

1 = C	5 = E	9 = G#
2 = C#	6 = F	10 = A
3 = D	7 = F#	11 = A#
4 = D#	8 = G	12 = B

lower limit = 1
upper limit = 13

The program produced this twelve-tone row:

10 = A, 3 = D, 4 = D#, 2 = C#, 9 = G#, 11 = A#, 12 = B, 7 = F#, 1 = C, 6 = F, 8 = G, 5 = E.

Using the row in its original form, this short piano composition was arranged:

FURTHER EXPLORATIONS

1. Use the program and one of the techniques outlined in this section to create a composition. Then select one series of notes to be used as a "thematic" melody repeated at intervals throughout the composition.
2. Research other harmonizing techniques to be used with a melody produced by the program.

REFERENCES/ INFORMATION

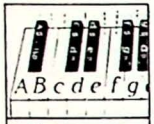
Reginald Smith Brindle, *Serial Composition* (Oxford: Oxford University Press, 1977).

William Christ, Richard DeLone, Vernon Kliewer, Lewis Rowell, and William Thomson, *Materials and Structure of Music I* (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1966).

Seppo Heikinheimo, *The Electronic Music of Karlheinz Stockhausen* (Helsinki: Suomen Musikkitieteellinen Seura, 1972).

William Thomson, *Introduction to Music as Structure* (Reading, Mass.: Addison-Wesley Publishing Company, 1971).

5 Pitch Frequency



OBJECTIVE

Each musical sound (pitch) can be represented as a wave, and the frequency of a pitch is determined by the number of cycles the wave completes in one second (cps = cycles per second). Since the intervals between the half tones in the scale are fixed, it is possible to calculate the frequency of any note if you know the frequency of a beginning, or tuning, note. These frequencies can then be used in synthesizer music.

The program listed here calculates these frequencies for the tempered scale.

FACTS/ THEORY

Throughout the history of music in Western civilization, several systems of musical tuning have been used — Pythagorean, meantone, just, and tempered. The tempered scale is the one most widely used today and is derived from a few simple premises:

1. The frequency relationship of the octave is 1:2.
2. There are twelve half steps in the octave.
3. The intervals between each of these half steps are exactly the same size.

The relationship between the frequencies of notes in the rising tempered scale can be algebraically expressed as

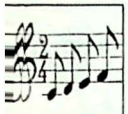
$$y = xk^n$$

where
- y = the frequency you want to identify,
- x = the frequency of the tuning note,
- k = a constant $= \sqrt[12]{2}$,
- n = the number of half steps between the two notes.

Similarly, to find the frequency of a note lower on the scale than the tuning note, the formula is

$$y = xk^{-n}.$$

The following program completes both these operations, depending on which subroutine you implement. You can easily use the program to calculate the frequencies of each note in the scale, rising or descending, or you can calculate the frequency of a particular note you want to identify, either above or below your tuning note.



THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [A] [STO] 1 [R/S]	Stores tuning note's frequency (x).
005	[STO] 02 12 [STO] 3 [R/S]	Stores the number of half-step intervals between note x and note y. Then stores number of half steps in scale in memory 3, which is used to control looping for calculation of entire scale's frequencies.
012	[2nd] [Lbl] [B] [RCL] 1 [×] [(] 2 [y^x] 12 [1/x] [)] [y^x] [RCL] 2 [=] [2nd] [Prt] [R/S]	Subroutine for rising scale. Calculates frequency of y.
030	[2nd] [Op] 22 [2nd] [Dsz] 3 [B] 0 [R/S]	Increases memory 2 (n) by 1 for calculation of frequency of next note in scale. Then subtracts 1 from memory 3 and checks to see if contents = 0. If not, loops back to [B] to calculate next frequency. When content reaches 0, displays 0 and stops.
037	[2nd] [Lbl] [C] [RCL] 1 [×] [(] 2 [y^x] 12 [1/x] [)] [y^x] [RCL] 2 [+/−] [=] [2nd] [Prt] [R/S]	Subroutine for descending scale.
056	[2nd] [Op] 22 [2nd] [Dsz] 3 [C] 0 [R/S]	Increases memory 2 by 1; checks to see if loop is required. If not, displays 0 and stops.
	[LRN] [RST]	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}		R_{03}	Used
R_{01}	x	R_{04}	
R_{02}	n		

Pitch Frequency

Using the Program

Procedure	Press	Display/Comments	
Enter frequency of tuning note (x).	x [A]	x	
To calculate frequencies of entire scale, enter 1.	1 [R/S]	12.	Shows entire number of half steps in scale.
To calculate entire rising scale:	[B]	y_1*	Frequency of note 1 half step *above* tuning note.
	[R/S]	y_2*	Note 2 half steps above tuning note.

Continue to press [R/S] until all twelve frequencies have been calculated. Display will show **0** when run is completed.

To calculate entire descending scale:	[C]	y_1*	Frequency of note 1 half step *below* tuning note.
	[R/S]	y_2*	

Continue to press [R/S] until all twelve frequencies are generated. Display shows 0 when run is completed.

OR		
Enter frequency of tuning note (x).	x [A]	x
To calculate frequency of one note only, enter number of half-step intervals between note x and note y.	n [R/S]	12.
To calculate one note only on rising scale:	[B]	y_n*
To calculate one note only on descending scale:	[C]	y_n*

*This value is printed if the PC-100A is used.

SAMPLE PROBLEM

1. Determine the frequency of each note in the octave A^1 to A^2, given that the tuning note A^1 has a frequency of 440 cps.

Procedure	Press	Display/Comments	
Enter program.		0	
Enter frequency of tuning note.	440 [A]	440.	
Enter 1.	1 [R/S]	12.	
Calculate frequencies.	[B]	466.1637615	A# or B♭
	[R/S]	493.8833013	B
	[R/S]	523.2511306	C
	[R/S]	554.365262	C# or D♭
	[R/S]	587.3295358	D
	[R/S]	622.2539674	D# or E♭
	[R/S]	659.2551138	E
	[R/S]	698.4564629	F
	[R/S]	739.9888454	F# or G♭
	[R/S]	783.990872	G
	[R/S]	830.6093952	G# or A♭
	[R/S]	880.	A^2
	[R/S]	0	

Note that the frequency of A^2 is exactly twice the frequency of A^1.

2. Calculate the frequency of F below A, given that the frequency of A is 440 cps.

Procedure	Press	Display/Comments
Enter frequency of A.	440 [A]	440.
Enter number of half-step intervals between A and F.	4 [R/S]	12.
Calculate frequency of F.	[C]	349.2282314

FURTHER EXPLORATIONS

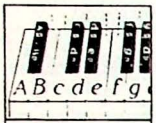
Design a program for determining the frequency of any pitch in one of the other systems of tuning, such as the Pythagorean system.

REFERENCES/ INFORMATION

John Backus, *The Acoustical Foundations of Music* (New York: W. W. Norton and Company, Inc., 1969).

Juan G. Roederer, *Introduction to the Physics and Psychophysics of Music* (New York: Springer-Verlag. 1973).

5 Harmonics



OBJECTIVE

Almost all sounds, including those produced by musical instruments, are actually combinations of pitches of various frequencies. These tones, called partials or harmonics, create the tone quality of the whole sound. The main frequency, called the fundamental, is the first harmonic.

The program in this section can be used to compute the frequencies of simple harmonics, given the frequency of the fundamental. These frequencies can then be used in synthesizers to create the tone quality you want.

FACTS/ THEORY

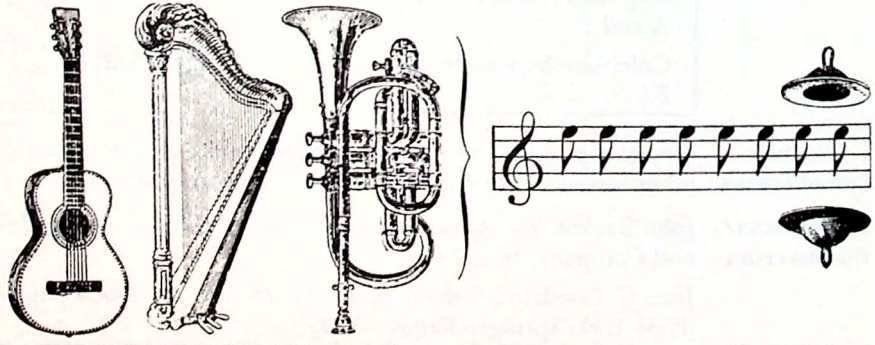
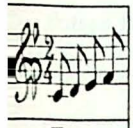
Partials are identified by number, and the number, frequency, and amplitude of the partials determine the tone quality of the sound. Theoretically, partials extend all the way to infinity; however, musicians usually concern themselves only with the first through the fourteenth partials. The frequencies above this point are usually inaudible to our ears.

A so-called pure tone — one without partials above the fundamental — is represented by a sine wave, and the sound can be compared to that produced by a french horn or flute, mellow and somewhat muted. On the other hand, if all the frequencies in a sound are amplified to equal levels, the result is what is called "white noise."

In musical tones that remain virtually unchanged for at least a few seconds, the partials are related to the fundamental in a very simple way: their frequencies are whole number multiples of the frequency of the fundamental pitch. The relationship may be expressed as

$$y = xn$$

where y = the frequency of the partial,
x = the frequency of the fundamental, and
n = the number of the partial.



THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 1 R/S	Stores frequency of fundamental.
005	2nd Lbl B STO 2 R/S	Stores number of harmonic.
010	2nd Lbl C RCL 1 × RCL 2 = 2nd Prt R/S	Calculates frequency of harmonic.
	LRN RST	Exits learn mode.

Note: Print command may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}
R_{01} x
R_{02} n
R_{03}

USING THE PROGRAM

Procedure	Press	Display/Comments
Enter frequency of fundamental.	x A	x
Enter number of partial.	n B	n
Calculate frequency of partial.	C	y°

°This value is printed if the PC-100A is used.

5 Harmonics

Sample Problem

What are the frequencies of the first, fifth, and seventh harmonics of a pitch whose fundamental is 261.63cps?

Procedure	Press	Display/Comments	
Enter program.		0	
Select 2-decimal display.	2nd fix 2	0.00	
Enter frequency of fundamental.	261.63 A	261.63	
Enter number of partial and calculate frequency.	1 B C	261.63	Frequency of 1st harmonic.
Enter next partial and calculate frequency.	5 B C	1308.15	Frequency of 5th harmonic.
Enter last partial and calculate frequency.	7 B C	1831.41	Frequency of 7th harmonic.

Further Explorations

Design a program for a synthesizer, using the frequencies generated by the program included here.

References/ Information

John Backus. *The Acoustical Foundations of Music* (New York: W. W. Norton and Company, Inc., 1969).

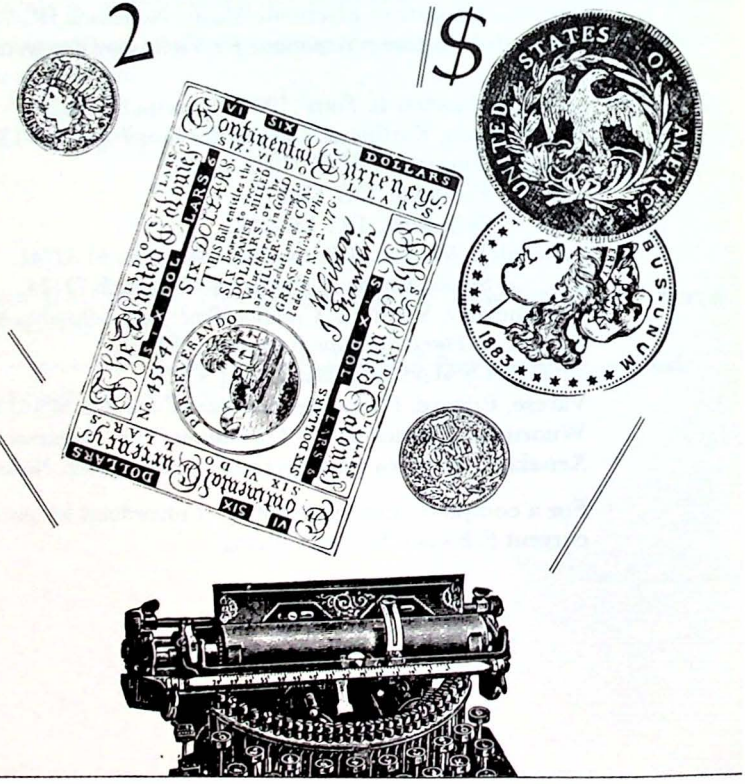
Juan G. Roederer, *Introduction to the Physics and Psychophysics of Music* (New York: Springer-Verlag, 1973).

Selected Listening Guide for Computer and Electronic Music

Babbitt, Milton. *Composition for Synthesizer* (1963). Columbia MS-6566.
_____. *Ensemble for Synthesizer* (1967). Columbia MS-7051.
Berio, Luciano. *Thema: Omaggio a Joyce* (1958). Turnabout 34177.
_____. *Visage* (1961). Turnabout 34046.
Cage, John. *Aria with Fontana Mix* (1958). Mainstream 5005.
_____. *Variations IV.* Everest 3230.
Cope, David. *Arena.* Orion ORS-75169.
_____. *K and Weeds.* (1971). Discant 1297.
Davidovsky, Marion. *Synchronisms 1, 2, and 3.* CRI S-204.
_____. *Synchronisms 4, 5, and 6.* Turnabout 34487.
Dodge, Charles. *Earth's Magnetic Field* (1970). Nonesuch 71250.
Hiller, L. and Baker. *Computer Cantata* (1963). CRI S-310.
Hiller, L. and John Cage. *HPSCHD* (1968). Nonesuch H-71224.
Hiller, Lejaren. *Machine Music* (1964). Turnabout 34536.
Kagel, Mauricio. *Acoustica* (1970). DC-2707059.
Ligeti, György. *Articulation* (1958). Wergo 60059.
Luening, Otto. *Synthesis* (1960). CRI S-219.
Nonesuch Guide to Electronic Music. Nonesuch HC-73018.
Randall, J. K. *Lyric Variations for Violin and Computer* (1967). Vanguard C-10057.
_____. *Quartets in Pairs* (1964). Nonesuch 71245.
Stockhausen, Karlheinz. *Gesang der Jünglinge.* DG-138811.
_____. *Hymnen.* DG-2707039.
_____. *Kontakte.* DG-138811.
_____. *Telemusik.* DG-137-012.
Subotnick, Morton. *4 Butterflies.* Columbia M-32741.
_____. *Silver Apples of the Moon.* Nonesuch 71174.
Ussachevsky, Vladimir. *Creation-Prologue.* Columbia MS-6566.
_____. *A Piece for Tape Recorder.* CRI-122.
_____. *Of Wood and Brass.* CRI S-227.
Varese, Edgard. *Poeme électronique.* Columbia MS-6146.
Wuorinen, Charles. *Times Encomium* (1969). Nonesuch 71225.
Xenakis, Iannis. Various electronic compositions. Nonesuch 71246.

For a complete up-to-date listing of recordings for purchase, check the current Schwann Record Catalog.

6 Business and Operations Research



Introduction

The daily operation of almost every business, whether large or small, requires that many decisions be made. Some of these decisions are financial in nature, some involve production or sales planning and forecasting, and all are vital to the on-going good health of the company. Today's handheld programmable calculators offer the modern businessman a powerful and exciting tool to use in making these decisions.

The examples shown in this chapter are just a few of the possible applications and were selected to show the versatility of the calculator. Included are the following topics:

- exploring an inventory-control model (EOQ),
- finding the present value, future value, and internal rate of return for a series of periodic unequal cash flows,
- calculating a capital-budgeting model, with the choice of straight-line, sum-of-the-years'-digits, or declining-balance depreciation,
- setting up a queuing model that reduces the cost of the service facility, and
- forecasting market shares (Markov Analysis).

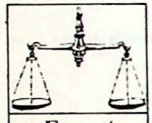
In each case, once the program is entered, the real value of the calculator becomes evident. You can easily change the input parameters and observe directly the effects of the changes. Manipulating the model in this way helps you determine optimum values for your situation and test for sensitivity in certain areas.

6 Inventory Control

OBJECTIVE

Inventory control plays an important part in the financial management of most businesses. Capital that could be used in other ways is tied up in holding inventories, and the related costs of storage and insurance increase with the size of the inventory; therefore, companies must try to find the most economical quantities to be held in inventory.

This section investigates the generalized inventory model known as the *Economic Order Quantity* (EOQ), and the program included here calculates the EOQ, the minimum inventory costs for a specified time period, and the number of times to order replacement stock during the period.



FACTS/ THEORY

The model we are using is based on the following formulas:

$$EOQ = \sqrt{\frac{2KL}{M}} + S$$

$$C_{min} = \sqrt{2KLM}$$

$$N = \frac{L}{EOQ}$$

where EOQ = the Economic Order Quantity in units.
C_{min} = the minimum inventory costs for the time period,
N = the number of times replacement stock should be ordered during the period,
K = the cost of placing and receiving each order,
L = the total number of units used during the period,
M = the cost of carrying one unit in inventory for the time period, and
S = the "safety" quantity of stock to be held in inventory.

"Safety" stock is that inventory held to minimize the risks or losses involved in running out of a crucial part or item. If sales would be lost or manufacturing production stopped because of a "stock-out," or if usage rate varies considerably and delivery time for replacement stock is long, a firm will usually require a fairly high level of safety stock. We won't discuss here the methods used to determine stock-out costs, but a full treatment of this subject is included in many production and operations research textbooks.

To use the following program, first enter the fixed cost of placing and receiving an order (K), the total (or estimated) number of units to be used, the cost of carrying one unit in inventory, and the safety stock quantity. You can then calculate the EOQ, the minimum inventory costs (C_{min}), and the number of times replacement stock should be ordered (N).

THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [A] [STO] 1 [R/S] [STO] 2 [R/S] [STO] 3 [R/S] [STO] 4 [R/S]	Stores K, L, M, and S.
014	[2nd] [Lbl] [B] 2 [×] [RCL] 1 [×] [RCL] 2 [÷] [STO] 5 [RCL] 3 [=] [√x] [+] [RCL] 4 [=] [STO] 6 [2nd] [Prt] [R/S]	Calculates EOQ.
038	[2nd] [Lbl] [C] [RCL] 5 [×] [RCL] 3 [=] [√x] [2nd] [Prt] [R/S]	Calculates C_{min}.
049	[2nd] [Lbl] [D] [RCL] 2 [÷] [RCL] 6 [=] [2nd] [Prt] [R/S]	Calculates N.
	[LRN] [RST]	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}		R_{04}	S
R_{01}	K	R_{05}	2KL
R_{02}	L	R_{06}	EOQ
R_{03}	M	R_{07}	

USING THE PROGRAM

Procedure	Press	Display/Comments
Clear memories.	[2nd] [CMs]	0.
Set decimal for desired accuracy.	[2nd] [Fix] n	
Enter data.	K [A]	K
	L [R/S]	L
	M [R/S]	M
	S [R/S]	S
Calculate EOQ.	[B]	EOQ°
Calculate C_{min}.	[C]	C_{min}°
Compute N.	[D]	N°

°This value is printed if the PC-100A is used.

6 Inventory Control

Sample Problem

A manufacturing company annually uses 30,000 units of a certain part. Because the part is essential to several production lines and delivery time on reorders is usually about two weeks, the company wants to maintain a safety stock of 1,000 units. Each order for replacement stock costs approximately $20, and the cost of carrying one unit in inventory for one year is estimated to be $0.12. What is the Economic Order Quantity? What are the minimum annual inventory costs, and how many times each year should replacement stock be ordered?

K = $20 M = $0.12
L = 30,000 units S = 1,000 units

Procedure	Press	Display/Comments
Enter program.		0
Set decimal at two places.	2nd fix 2	0.00
Enter data.		
K	20 A	20.00
L	30000 R/S	30000.00
M	.12 R/S	0.12
S	1000 R/S	1000.00
Calculate EOQ.	B	4162.28
Calculate C_{min}.	C	379.47
Calculate N.	D	7.21

The Economic Order Quantity is about 4,162 units. Minimum inventory costs are $379.47 per year, and orders should be placed approximately seven times a year.

Further Explorations

Test the model for sensitivity to slight changes in K, L, and M. For instance, what happens in the example listed above if M is increased to $0.15? (*Answer:* EOQ becomes 3828 units, C_{min} is $424.26, and N is 8.)

References/ Information

J. Fred Weston and Eugene F. Brigham, *Essentials of Managerial Finance,* 2nd ed. (New York: Holt, Rinehart and Winston, Inc., 1968), pp. 310-314.

Variable Periodic Cash Flows and Internal Rate of Return

OBJECTIVE

FACTS/ THEORY

Very common in business are annuity-like situations in which the periodic cash flows are unequal amounts. The program included here calculates present value (PV), future value (FV), and internal rate of return (IRR) for a series of variable periodic cash flows.

Like annuity payments, variable cash flows can occur either at the end of the period or at the beginning of the period. For calculating PV and FV the formulas are:

Payments at end of period

$$PV = \sum_{1}^{n} CF_k\,(1+i)^{-k}$$

$$FV = \left[\sum_{1}^{n} CF_k\,(1+i)^{-k}\right](1+i)^{n}$$

Payments at beginning of period

$$PV = \left[\sum_{1}^{n} CF_k\,(1+i)^{-k}\right](1+i)$$

$$FV = \left[\sum_{1}^{n} CF_k\,(1+i)^{-k}\right](1+i)^{n+1}$$

where CF_k = cash flow at period k (k = 1 through n),
i = periodic interest rate expressed as a decimal,
k = intermediate period from 1 to n, and
n = total number of compounding periods.

Note that cash flows occurring at the beginning of the period are compounded for n + 1 periods, rather than n periods.

To find IRR it is necessary to use an iterative solution, and we're using here the Newton-Raphson method to approximate the interest rate i (or IRR):

$$i_{x+1} = i_x - \frac{f(i)}{f'(i)}$$

$$f(i) = \left[\sum_{k=1}^{n} CF_k\,(1+i)^{-k}\right] - PV$$

$$f'(i) = -n\left[\sum_{k=1}^{n} CF_k\,(1+i)^{-(k+1)}\right]$$

(For a detailed explanation of the Newton-Raphson method, see Chapter 3, "Approximating Roots of a Function.")

6

Variable Periodic Cash Flows and Internal Rate of Return

In the program listed below we enter 0 as our first "guess" (i) and instruct the program to loop for 10 iterations, or until the change in i is less than .00001.

You may want to change the initial i, the number of iterations, and/or the accuracy limit to fit the requirements of your particular situation. These changes can be made easily by minor editing of the program. In some cases where negative cash flows are present, it is possible that there is more than one root to the equation. Our program, in general, finds the root (i) closest to zero that satisfies the conditions you enter. If you suspect that there may be multiple roots, you may want to enter ± 0.9 as an initial "guess."

Given the interest rate i, the program solves for PV or FV for a series of periodic cash flows occurring either at the end or at the beginning of the period. Given the cash flows and either PV or FV, the program also finds IRR. Some of these calculations require special entry for certain cash flows, and these special cases are noted in *Using the Program* below.

The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl 2nd B' 2nd Op 20 RCL 2nd Ind 6 × (1 + RCL 8) y^x RCL 0 +/− = SUM 4 × RCL 0 = ÷ (1 + RCL 8) = +/− SUM 5 2nd Op 26 2nd Dsz 1 2nd B' RCL 4 − RCL 7 = INV SBR	Calculates $\Sigma\, CF_k(1+i)^{-k}$ and $\Sigma\, -kCF_k\,(1+i)^{-(k+1)}$ (summations for PV and IRR calculations).
047	2nd Lbl 2nd E' RCL 8 2nd Prt R/S	Recalls new IRR if difference between old IRR_{est} and new IRR_{est} is less than .00001.
053	2nd Lbl E 11 STO 6 RCL 2 STO 1 CLR STO 0 STO 4 STO 5 INV SBR	Initializes memory registers for new summation.

Step #	Program Keystrokes	Purpose/Comments
071	2nd Lbl 2nd A' CLR STO 08 10 STO 9 2nd Lbl EE .00001 x:t E 2nd B' ÷ RCL 5 = +/- + RCL 8 STO 3 = STO 8 − RCL 3 = 2nd Lx 1 INV 2nd x=t 2nd E' E 2nd Dsz 9 EE RCL 8 2nd Prt R/S	Stores 0 as first "guess" for IRR routine, 10 as number of iterations, and .00001 as accuracy limit. When change between new and old $IRR_{est} < .00001$, exits loop and displays new estimate as IRR.
120	2nd Lbl A STO 2nd Ind 6 2nd Op 26 2nd Op 22 2nd Op 21 R/S	Stores CF_k (memory 6 used for indirect storage and recall) and increases memory registers 1 and 2 by 1.
131	2nd Lbl B STO 7 R/S	Stores PV for IRR routine.
136	2nd Lbl C STO 8 E 2nd B' RCL 4 2nd Prt R/S	Stores i and computes PV (payments at end of period).
146	2nd Lbl 2nd C' STO 8 E 2nd B' RCL 4 × (1 + RCL 8) = 2nd Prt R/S	Stores i and computes PV (payments at beginning of period).
164	2nd Lbl D STO 8 E 2nd B' RCL 4 × (1 + RCL 8) y^x RCL 2 = 2nd Prt R/S	Stores i and computes FV (payments at end of period).
185	2nd Lbl 2nd D' STO 8 E 2nd B' RCL 4 × (1 + RCL 8) y^x (RCL 2 + 1) = 2nd Prt R/S	Stores i and computes FV (payments at beginning of period).
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

6

Variable Periodic Cash Flows and Internal Rate of Return

Register Contents

R_{00}	Used	R_{05}	$\Sigma -k(CF_k)(1+i)^{-(k+1)}$	R_{10}	
R_{01}	Used	R_{06}	Used	R_{11}	Remaining
R_{02}	n	R_{07}	PV	R_{12}	registers
R_{03}	IRR_{old}	R_{08}	i	R_{13}	used as needed
R_{04}	$\Sigma CF_k(i+i)^{-k}$	R_{09}	Used	R_{14}	for CF's.

Using the Program

For Payments at End of Period

Procedure	Press	Display/Comments
Clear memories and set decimal for desired accuracy.	2nd CMs 2nd Fix n	0.
To find PV or FV:		
Initialize.	E	0.
Enter cash flows 1 through n.	CF_1 A	CF_1
	...	...
	CF_n A	CF_n
Enter i as a decimal and calculate PV or FV.	i C	PV°
	OR	
	i D	FV°
To find IRR, PV known:		
Initialize.	E	0.
Enter cash flows 1 through n.	CF_1 A	CF_1
	...	...
	CF_n A	CF_n
Enter PV.	PV B	0
Calculate IRR.	2nd A'	IRR° Expressed as a decimal.
To find IRR, FV known:		
Initialize.	E	0.
Enter cash flows 1 through n-1.	CF_1 A	CF_1
	...	...
	CF_{n-1} A	CF_{n-1}
Enter $CF_n - FV$ as last cash flow.	$CF_n - FV$ = A	CF_n $(CF_n - FV = CF_n)$
Enter 0 as PV.	0 B	0
Compute IRR.	2nd A'	IRR°

°This value is printed if the PC-100A is used.

For Payments at Beginning of Period

Procedure	Press	Display/Comments
Clear memories and set decimal for desired accuracy.	2nd CMs 2nd Fix n	0.
To find PV or FV: Initialize.	E	0.
Enter cash flows 1 through n.	CF_1 A ••• CF_n A	CF_1 ••• CF_n
Enter i as a decimal and calculate PV or FV.	i 2nd C' *OR* i 2nd D'	PV° FV°
To find IRR, PV known: Initialize.	E	0.
Enter cash flows 2 through n.	CF_2 A ••• CF_n A	CF_2 ••• CF_n
Enter $PV - CF_1$.	PV − CF_1 = B	0
Calculate IRR.	2nd A'	IRR°
To calculate IRR, FV known:		
Initialize.	E	0
Enter cash flows 1 through n.	CF_1 A ••• CF_n A	CF_1 ••• CF_n
Enter $-FV$ as final cash flow.	FV +/− A	$-FV$
Enter 0 as PV.	0 B	0
Compute IRR.	2nd A'	IRR°

°This value is printed if the PC-100A is used.

6

Variable Periodic Cash Flows and Internal Rate of Return

Sample Problem

Calculation of IRR, FV known (cash flows at end of period)

Your company has made the following deposits (at the end of each month) in an investment account:

Month	1	2	3	4	5
Deposit	$500	$800	$1000	$1200	$1500

At the end of the fifth month, the amount in the account is $5564.33. What return rate are you earning?

Procedure	Press	Display/Comments
Enter program.		0
Set decimal at 4 places.	2nd fix 4	0.0000
Initialize.	E	0.0000
Enter all but the last cash flow.	500 A	500.0000 CF_1
	800 A	800.0000 CF_2
	1000 A	1000.0000 CF_3
	1200 A	1200.0000 CF_4
Subtract FV from CF_5 and enter as last cash flow.	1500 −	1500.0000
	5564.33 = A	−4064.3300 CF_5
Enter 0 as PV.	0 B	0
Compute IRR.	2nd A'	0.0700 IRR

The account is earning 7%.

Calculation of IRR, PV known (cash flows at beginning of period)

A manufacturer will either sell a machine for $50,000 or lease it for five years, with the lease payments to be made at the beginning of each year. The lease payments would be:

Year	1	2	3	4	5
Payment	$14,500	$13,500	$12,500	$11,000	$10,000

What return rate will the lease situation earn?

Procedure	Press	Display/Comments
Initialize.	E	0.0000
Enter CF_2 through CF_5.	13500 A	13500.0000
	12500 A	12500.0000
	11000 A	11000.0000
	10000 A	10000.0000
Subtract CF_1 from cost of machine and enter as PV.	50000 −	50000.0000
	14500 = B	35500.0000 This is the cost of the project to the manufacturer; i.e., the money he does not earn if he leases, rather than sells, the machine.
Calculate IRR.	2nd A'	0.1299

The lease arrangement earns almost 13% for the manufacturer.

FURTHER EXPLORATIONS

The program above assumes that the compounding period and the cash flow period are the same. What happens if they are different? For example, consider a situation with monthly cash flows and quarterly compounding. To calculate PV or FV for this case, you must convert the interest rate before entering it into the program. The interest rate per cash flow period (γ) is given by:

$$\gamma = \left[(1+i)^{m/k}\right] - 1$$

where i is the interest rate per compounding period, m is the number of compounding periods per year, and k is the number of payments or cash flows per year.

To calculate IRR where the cash flow or payment period differs from the compounding period, you simply convert the calculated IRR with the following equation:

$$i = \left[(1+\gamma)^{k/m}\right] - 1$$

Here γ is the calculated IRR per payment period, and i is the equivalent IRR per compounding period.

Change the program to perform these calculations.

REFERENCES/ INFORMATION

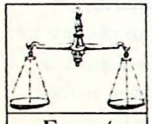
Roger F. Farish *et al, Calculator Analysis for Business and Finance* (Dallas: Texas Instruments Incorporated, 1977), pp. 4-23—4-30.

6 Capital Budgeting

OBJECTIVE

Capital budgeting is a complex process that is essential to sound financial management in business. It involves the planning and evaluation of any capital outlay with expected returns extending beyond one year's time – the acquisition of land or buildings, for example, or the purchase of new equipment. Because expenditures of this type mean that a certain amount of capital must be tied up or committed initially, with returns spread over several years, these projects must be examined carefully for their potential profitability.

The program listed here allows you to find the present value, after taxes and depreciation, of a series of cash flows created by the acquisition of an asset. The result can be used as a measure of the profitability of the project.



FACTS/ THEORY

Before capital budgeting can begin, the returns (cash inflows, savings, etc.) of the project must be estimated. Obviously the forecast of these returns is critical, since overestimation or underestimation can give a completely false picture of the project's profitability. Once the returns have been estimated, the next step is to calculate the present value of the projected returns and then to compare this to the cost of the project.

This generalized capital budgeting model uses the following expressions:

$$NCF_n = CF_n(1-T_1)+D_n(T_1)$$

$$ACF = P-[(P-NBV_k)\times T_2]$$

$$NBV = RDV+SAL$$

$$PV = \left[\sum_{1}^{k} NCF_n(1+R)^{-n}\right]+ACF(1+R)^{-k}$$

where NCF_n = net cash flow at end of period n after taxes and depreciatio
CF_n = cash flow at end of period n before taxes and depreciation,
T_1 = average tax rate on annual income,
D_n = depreciation during period n,
ACF = after-tax cash flow from selling asset at end of period k,
P = gross proceeds (cash received) from sale of asset,
NBV = net book value of asset at end of period k,
T_2 = tax rate on gain or loss from sale of asset,
RDV = remaining depreciable value at end of period k,
SAL = salvage value for depreciation,
PV = present value of cash returns or savings,
R = required rate of return, expressed as a decimal,
k = decision period or period at end of which the asset is resold, and
n = any intermediate period between purchase of asset and its resale.

The following program solves these expressions and calculates the present value of the total after-tax cash inflows or savings generated by the asset. The cost of the asset is then subtracted from the computed present value to find the net present value of the project. If the net present value is negative, the project does not meet the company's required return rate. If zero or positive, however, the project meets or exceeds the company's required rate of return.

Three types of depreciation are offered by the program, depending on which subroutine you select. Subroutine [C] calculates NCF_n using straight-line depreciation; subroutine [D] uses the declining balance method, and subroutine [E] the sum-of-the-years'-digits method.

A note about Declining Balance Depreciation: When this method is used, two considerations are very important — one, the asset can't be depreciated (for tax purposes) below its salvage value; and, two, if the asset will not be fully depreciated down to salvage value by (or before) the end of its life, the company should switch over at some point to the straight-line depreciation method to maximize the tax advantage. (The crossover point is the period when the straight-line depreciation amount exceeds the declining balance amount.) Safeguards are "built into" the declining balance subroutine to meet these conditions.

If you are using the TI Programmable 58, you will need to repartition the calculator since the program has more than 240 program steps. See pages V-42 and V-43 in *Personal Programming* for a discussion of partitioning.

THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [A] [STO] 2 [STO] 11 [STO] 13 [R/S]	Stores cost of asset.
009	[STO] 3 [INV] [SUM] 11 [INV] [SUM] 13 [R/S]	Stores salvage value of asset for depreciation.
018	[STO] 4 [+] 1 [=] [×] [RCL] 4 [÷] 2 [=] [STO] 15 [R/S]	Stores life of asset and sum-of-the-years'-digits for SOYD depreciation.
032	[STO] 9 [CLR] [STO] 0 [R/S]	Stores resale value of asset and clears counting register.
038	[2nd] [Lbl] [B] [STO] 5 [R/S]	Stores income tax rate (T_1).

Step #	Program Keystrokes	Purpose/Comments
043	STO 6 R/S	Stores gains/loss tax rate on sale of asset (T_2).
046	+ 1 = STO 7 R/S	Stores 1 + required rate of return (R).
052	STO 8 R/S	Stores factor for declining balance depreciation. *Note:* Data *must* be entered in the order shown above.
055	2nd Lbl C STO 1 RCL 11 ÷ RCL 4 = STO 10 INV SUM 13	*Straight-Line Depreciation:* Calculates depreciation for period n and remaining depreciable value (RDV).
070	2nd Lbl 2nd B' 2nd CP 20 RCL 1 × (1 − RCL 5 = + RCL 5 × RCL 10 = × RCL 7 y^x RCL 0 +/− = SUM 12 RCL 12 2nd Prt R/S	Calculates PV of net cash flow. Sums and displays PV at end of period n.
104	2nd Lbl D STO 1 2nd CP RCL 13 2nd x≥t 2nd D'	*Declining Balance Depreciation:* Checks to see if RDV = 0. If so, transfers to 2nd D'.
113	÷ (RCL 4 − RCL 0 = STO 10 2nd Mfg 1 2nd C' x≥t	Calculates modified straight-line depreciation for later comparison to declining balance depreciation.
127	RCL 2 × (1 − RCL 8 ÷ RCL 4) y^x RCL 0 = × RCL 8 ÷ RCL 4 = STO 14 INV 2nd x≥t 2nd C' 2nd CP +/− + RCL 13 = INV 2nd x≥t 2nd E'	Calculates declining balance depreciation for period and compares to SL amount. If DB < SL (crossover point), transfers to 2nd C'. Then checks to see if DB depreciation will depreciate asset below salvage. If so, transfers to 2nd E'.

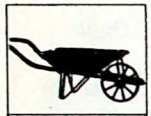
Step #	Program Keystrokes	Purpose/Comments
164	RCL 14 STO 10 INV SUM 13 2nd B'	If all tests default, DB depreciation expense is allowed, and program transfers to 2nd B' to find PV_n.
172	2nd Lbl E STO 1 RCL 4 − RCL 0 = ÷ RCL 15 = X RCL 11 = STO 10 INV SUM 13 2nd B'	*Sum-of-the-Years'-Digits Depreciation:* Calculates depreciation for period n and RDV. Then loops to 2nd B' to calculate PV_n.
196	2nd Lbl 2nd A' RCL 9 − (RCL 13 + RCL 3 = X RCL 6 = +/− + RCL 9 = X RCL 7 y^x RCL 0 +/− = + RCL 12 = 2nd Prt R/S − RCL 2 = 2nd Prt R/S	Computes after-tax cash flow (ACF) from selling asset, and calculates and displays PV of project. Then subtracts cost of asset to find NPV of project.
237	2nd Lbl 2nd C' 2nd St flg 1 RCL 10 INV SUM 13 2nd B'	Sets flag 1 to indicate that SL depreciation is preferred, rather than DB depreciation, for remainder of asset's life. Finds RDV using SL depreciation and loops to 2nd B' to calculate PV_n.
247	2nd Lbl 2nd D' CLR STO 10 2nd B'	Asset is fully depreciated (RDV = 0). Enters 0 as depreciation for period n and loops to 2nd B'.
253	2nd Lbl 2nd E' RCL 13 STO 10 INV SUM 13 2nd B'	Enters RDV as depreciation for period n and loops to 2nd B'.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

6 Capital Budgeting

Register Contents

R_{00}	Year counter	R_{05}	T_1	R_{10}	Dep_n
R_{01}	CF_n	R_{06}	T_2	R_{11}	Cost − salvage
R_{02}	Cost of asset	R_{07}	R + 1	R_{12}	NPV_n
R_{03}	Salvage value	R_{08}	DB factor	R_{13}	RDV
R_{04}	Life of asset	R_{09}	Resale value	R_{14}	Used
				R_{15}	SOYD



Using the Program

Procedure	Press	Display/Comments
Enter program.		**0**
Select two-place decimal setting.	2nd fix 2	**0.00**
Enter data.		
Cost of asset.	A	**Cost**
Salvage value.	R/S	**Salvage**
Life of asset.	R/S	**SOYD**
Resale value.	R/S	**0.00** (Year-counting register cleared.)
Income tax rate (T_1) as a decimal.	B	**T_1**
Gain/loss tax rate (T_2) as a decimal.	R/S	**T_2**
Required rate of return (R) as a decimal.	R/S	**(1 + R)**
DB factor.	R/S	**Factor** This step may be omitted if you are using SL or SOYD depreciation.
Enter CF_1 and press C, D, or E, depending upon which kind of depreciation you are using.	C	**NPV_1** ° With straight-line depreciation.
	or	
	D	**NPV_1** ° With declining balance depreciation.
	or	
	E	**NPV_1** ° With sum-of-the-years'-digits depreciation.

Procedure	Press	Display/Comments
Repeat for each cash flow. When all cash flows have been entered, calculate PV of project.	2nd A'	PV°
Compare with cost of asset or project.	R/S	NPV of project°

°This value is printed if the PC-100A is used.

SAMPLE PROBLEM

A company is evaluating a project involving the purchase of a new machine. The machine costs $50,000 and has a life of ten years, with a salvage value of $7,000 for depreciation purposes. At the end of the fifth year, however, the machine can be sold for $25,000. The tax rate on gain/loss will be 25%.

The company uses double declining balance depreciation (a factor of 2) for its assets and expects a 15% rate of return after tax on its projects. Taxes on income have averaged 40% in the past. According to best estimates, the project should create the following annual cash flows before depreciation and taxes:

Year	1	2	3	4	5
CF	3000	10000	15000	25000	35000

Will the project be profitable? In other words, will the present value of the project's returns exceed its initial cost?

Procedure	Press	Display/Comments	
Enter program.		0	
Clear memories and flags and select two-decimal accuracy.	2nd CMS RST 2nd Fix 2	0. 0.00	
Enter data.			
Cost of machine.	50000 A	50000.00	
Salvage value.	7000 R/S	7000.00	
Life of machine.	10 R/S	55.00	Sum-of-the-years'-digits.
Resale value.	25000 R/S	0.00	Counting register cleared.
Income tax rate.	.4 B	0.40	T_1
Gain/loss tax rate.	.25 R/S	0.25	T_2
Required return rate.	.15 R/S	1.15	(R + 1)
DB factor.	2 R/S	2.00	
Enter first year's cash flow (CF_1) and calculate PV at end of period 1 with DB depreciation.	3000 D	5043.48	PV at end of period 1.
Enter second year's cash flow (CF_2) and calculate PV at end of period 2.	10000 D	12000.00	PV at end of period 2.
Enter remaining cash flows:			
CF_3.	15000 D	19600.89	PV at end of period 3.
CF_4.	25000 D	29348.14	PV at end of period 4.
CF_5.	35000 D	40603.42	PV at end of period 5.

Procedure	Press	Display/Comments
Now calculate entire PV of the project.	2nd A'	51961.92 $\left[\sum_{1}^{k} NCF(1+R)^{-n}\right] + [ACF\,(1+R)^{-k}]$
Compare to cost of machine.	R/S	1961.92 NPV of project.

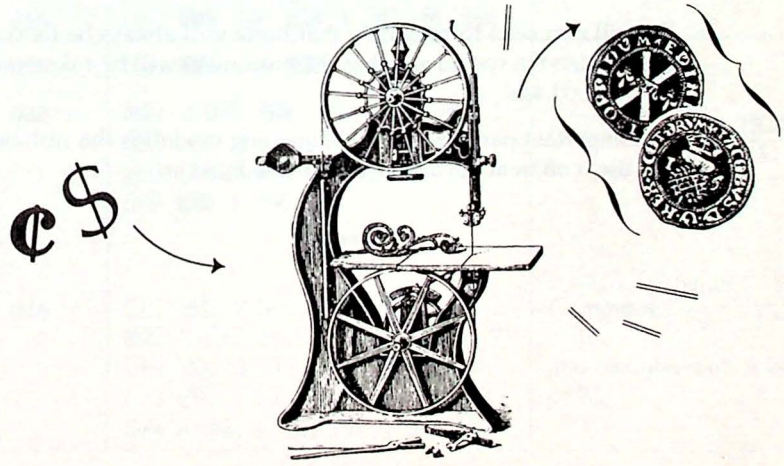
Since the net present value is positive, the project's return rate will exceed the rate of return required by the company. The project is profitable.

FURTHER EXPLORATIONS

1. Suppose the company had used straight-line or sum-of-the-years'-digits depreciation, rather than declining balance. Would the net present value of the project be affected?
2. Test the sensitivity of the model to slight changes in T_1, T_2, or R.

REFERENCES/ INFORMATION

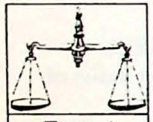
J. Fred Weston and Eugene F. Brigham, *Essentials of Managerial Finance*, 2nd ed. (New York: Holt, Rinehart and Winston, Inc., 1968), chapter 8. See also chapter 9 for a discussion of risk in capital budgeting procedures.



6 Queuing

OBJECTIVE

In many organizations there are processes which generate waiting lines or queues. Queues form when someone must wait for service because the servicing facility is temporarily unable to provide that service. Industry typically pays for both the lost time of persons waiting in line as well as the wages of the people who provide the service. Queuing theory provides a way for management to arrive at the lowest total cost by determining how many persons to hire for the servicing facility. This program handles the multiple server case with arrivals Poisson distributed and an exponential service time distribution. Although this is only one of several models, it is hoped that it will illustrate some of the potential of the use of the programmable calculator in this area of study.



FACTS/ THEORY

The terms used in this problem are:

N	– number of servers
μ	– average number of customers served per time period
λ	– average number of customers arriving per time period
P_0	– probability of nobody in the system
Cw	– cost per time period of customer waiting time
Cs	– cost per time period for one service line
TC	– total cost for operating service facility
Lq	– expected number of customers waiting in queue
L	– expected number of customers in the system (those being served plus those waiting for service)
Wq	– average time spent waiting to be served
W	– average time spent in the system (includes time spent being served and waiting for service)

We will assume in this example that there will always be more customers coming into the system and that the customers will be taken on a first-come, first-served basis.

One important parameter used in queuing models is the utilization factor. We will use it often in the model, and it is defined as:

$$\rho = \frac{\lambda}{N\mu}$$

For this particular model, with Poisson arrivals and exponential service times, the following equations are used.

$$P_0 = \left[\sum_{k=0}^{N-1} \frac{(N\rho)^k}{k!} + \frac{(N\rho)^N}{N!} \left(\frac{1}{1-\rho} \right) \right]^{-1}$$

$$Lq = \frac{N^N \rho^{N+1}}{N!(1-\rho)^2} P_0$$

$$Wq = \frac{Lq}{\lambda}$$

$$W = Wq + \frac{1}{\mu}$$

$$L = \lambda W$$

$$TC = NCw + Cs\lambda Wq$$

The following program allows you to enter N, λ, μ, Cw, and Cs, and then computes the remaining values. By varying the number of servers, you can minimize the desired values to analyze particular queuing situations.

THE PROGRAM

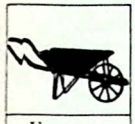
Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [2nd] [A'] [STO] 0 [2nd] [CP] [2nd] [x=t] [2nd] [C']	Subroutine to compute factorials.
007	[2nd] [Lbl] [2nd] [B'] [RCL] 0 [×] [2nd] [Dsz] 0 [2nd] [B']	
015	[2nd] [Lbl] [2nd] [C'] 1 [=] [INV] [SBR]	
020	[2nd] [Lbl] [2nd] [D']	Calculates P_0.
022	[RCL] 2 [2nd] [A'] [STO] 3 [(] [RCL] 7 [×] [RCL] 4 [)] [y^x] [RCL] 2 [÷] [RCL] 3 [=] [SUM] 8 [2nd] [Op] 32 [2nd] [Dsz] 1 [2nd] [D']	Computes $\sum_{K=0}^{N-1} \frac{(N\rho)^k}{k!}$
048	[(] [RCL] 7 [2nd] [A'] [STO] 3 [(] [RCL] 7 [×] [RCL] 4 [)] [y^x] [RCL] 7 [÷] [RCL] 3 [)] [×] [(] [(] 1 [−] [RCL] 4 [)] [1/x] [)] [=] [SUM] 8 [RCL] 8 [1/x] [STO] 8 [INV] [SBR]	Computes $\frac{(N\rho)^N}{N!} \left(\frac{1}{1-\rho} \right)$ and completes calculation of P_0.

Step #	Program Keystrokes	Purpose/Comments
087	2nd Lbl A STO 7 STO 1 − 1 = STO 2 R/S STO 6 R/S STO 5 ÷ (RCL 7 × RCL 6) = STO 4 R/S	Stores N, N − 1, μ, and λ. Computes ρ.
116	2nd Lbl B STO 13 R/S STO 14 R/S	Stores Cw and Cs.
124	2nd Lbl C	
126	2nd D' RCL 7 2nd A' × (1 − RCL 4) x^2 = STO 3 RCL 7 y^x RCL 7 × RCL 4 y^x (RCL 7 + 1) ÷ RCL 3 × RCL 8 = STO 09	Calls subroutine D' to compute P_0. Then completes calculation of Lq.
165	÷ RCL 5 = STO 10	Computes Wq.
171	+ RCL 6 1/x = STO 12	Computes W.
178	× RCL 5 = STO 11	Computes L.
184	RCL 13 × RCL 5 × RCL 10 + (RCL 7 × RCL 14) = STO 15 2nd Prt R/S	Computes TC.
205	2nd Lbl D RCL 9 2nd Prt R/S RCL 10 2nd Prt R/S RCL 11 2nd Prt R/S RCL 12 2nd Prt R/S	 Displays Lq. Displays Wq. Displays L. Displays W.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}	Used	R_{06}	μ	R_{11}	L
R_{01}	Used	R_{07}	N	R_{12}	W
R_{02}	N − 1	R_{08}	P_0	R_{13}	Cw
R_{03}	Used	R_{09}	Lq	R_{14}	Cs
R_{04}	ρ	R_{10}	Wq	R_{15}	TC
R_{05}	λ				



USING THE PROGRAM

Procedure	Press	Display/Comments
Fix decimal for desired accuracy.	2nd Fix n	
Enter N.	A	N − 1
Enter μ.	R/S	μ
Enter λ.	R/S	ρ
Enter Cw.	B	Cw
Enter Cs.	R/S	Cs
Begin calculations.	C	TC°
Display results.	D	Lq°
	R/S	Wq°
	R/S	L°
	R/S	W°

°This value is printed if the PC-100A is used.

SAMPLE PROBLEM

The Ace Machine Company uses forklift trucks to move pieces of machinery from one part of the factory to other parts for further work. If a worker is idle, waiting for a forklift to bring him a piece of machinery, he is still paid $7.00 per hour for his time. The company currently uses two forklifts and has determined it costs $12.00 per hour to operate each one. Requests for a forklift's service follow a Poisson distribution with an average of ten calls per hour. It takes a forklift an average of four minutes per request to move a piece of machinery. The service time is exponentially distributed. Use the program to determine if the firm should purchase an additional forklift.

Procedure	Press	Display/Comments
Enter program.		0
Fix decimal for desired accuracy. (We'll use 2.)	2nd Fix 2	0.00
Enter N.	2 A	1.00 N − 1
Enter μ.	15 R/S	15.00 μ
Enter λ.	10 R/S	0.33 ρ
Enter Cw.	7 B	7.00 Cw
Enter Cs.	12 R/S	12.00 Cs
Begin calculations for two forklifts.	C	24.58 TC

Procedure	Press	Display/Comments
Display results.	D	0.08 Lq
	R/S	0.01 Wq
	R/S	0.75 L
	R/S	0.08 W
Clear memories to begin next calculation.	CLR 2nd CMs	0.00
Enter N.	3 A	2.00 N-1
Enter rest of data, the same as above.	15 R/S	15.00 μ
	10 R/S	0.22 ρ
	7 B	7.00 Cw
	12 R/S	12.00 Cs
Begin calculations for three forklifts.	C	36.07 TC
Display results.	D	0.01 Lq
	R/S	0.00 Wq
	R/S	0.68 L
	R/S	0.07 W

It would cost the company more per hour to operate three forklifts. Thus, at the present time the company should not purchase a third forklift.

FURTHER EXPLORATIONS

1. Write a program that will find L, Lq, W, and Wq for a single server model with Poisson arrivals and either an exponential service time distribution or a constant service time.
2. Suppose we have a queue where all customers must pass through multiple servers in a line. Write a program to compute L, Lq, W, and Wq for this type of queue.

REFERENCES/ INFORMATION

R. I. Levin and C. A. Kirkpatrick, *Quantitative Approaches to Management* (New York: McGraw-Hill Book Company, 1965), chapter 15.

J. E. Ullmann, *Theory and Problems of Quantitative Methods in Management*, Schaum's Outline Series (New York: McGraw-Hill Book Company, 1976), chapter 19.

Markov Analysis

OBJECTIVE

Markov analysis is a method of analyzing the current movement of a variable in order to predict the future movement of that variable. As a tool for business management, Markov analysis is most often used as a marketing aid in predicting consumers' brand loyalty. The program here is designed to help you perform the matrix arithmetic involved in doing a Markov analysis.

FACTS/ THEORY

To demonstrate the basic application of Markov analysis, we'll use an example involving three companies (the process is the same regardless of the number of companies used). We'll assume that two pieces of data are known: the number of customers for each company and the company from which new customers were obtained. To simplify the mathematics involved, we will also assume that no new customers enter and no old customers leave the market during the period we wish to analyze.

Let's use the three companies X, Y, and Z for our analysis. Suppose we have the following records showing the gain and loss of customers each company experienced during the month.

		Gains			*Losses*			
Company	*March 1 Customers*	*From X*	*From Y*	*From Z*	*To X*	*To Y*	*To Z*	*April 1 Customers*
X	300	0	30	10	0	10	20	310
Y	500	10	0	25	30	0	15	490
Z	200	20	15	0	10	25	0	200

To use the Markov analysis, we will have to compute *transition probabilities:* the probability that a company will retain its customers, the probability that a company will gain new customers from another company, and the probability that a company will lose customers to another company.

These transition probabilities are usually presented in a transition matrix. The standard form of the matrix expresses the retention and loss by the rows, and retention and gain by the columns. Thus, for our example, the transition matrix is shown below.

		Retention and loss →			
	X	Y	Z		
X	270/300 = .900	10/300 = .033	20/300 = .067	*Retention*	
Y	30/500 = .060	455/500 = .910	15/500 = .030	*and*	
Z	10/200 = .050	25/200 = .125	165/200 = .825	*gain*	↓

6 Markov Analysis

In this problem we will use a first-order Markov process. This means that we assume that a purchase decision next month depends solely on the customers' choices this month. Studies have shown that first-order processes are reliable predictors of the future, especially when the transition matrix remains fairly stable.

We know that the April 1 market shares for each company are:

$$X = 310/1000 = .31$$
$$Y = 490/1000 = .49$$
$$Z = 200/1000 = .20$$

To calculate the expected market share for each company on May 1, we set up the April 1 market shares as a matrix and multiply this matrix by the transition matrix. This computation is shown below for our example.

April 1 market shares				*Transition matrix*				*Probable May 1 market shares*		
(.31	.49	.20)	×	.900	.033	.067	=	(.318	.481	.201)
				.060	.910	.030				
				.050	.125	.825				

To calculate the probable June 1 market shares, you simply multiply the probable May 1 market shares by the transition matrix.

It is reasonable to assume that, if none of the companies in the market takes any action to alter the transition matrix, then a state of equilibrium might be reached some time in the future. We can find the equilibrium market shares for our example by solving the following simultaneous equations:

$$X = .900\,X + .033\,Y + .083\,Z$$
$$Y = .060\,X + .910\,Y + .030\,Z$$
$$Z = .050\,X + .125\,Y + .825\,Z$$
$$1 = X + Y + Z$$

Since we have four equations and only three unknowns, we may drop one equation. Thus this system can be rewritten in the normal form as:

$$-.100\,X + .003\,Y + .083\,Z = 0$$
$$.060\,X - .090\,Y + .030\,Z = 0$$
$$X + Y + Z = 1$$

By solving this system of equations we may find the equilibrium market shares.

The following program will compute the probable future market shares and the equilibrium market shares for a three-by-three transition matrix. The program uses Master Library program ML-02 to solve the simultaneous equations.

The EProgram

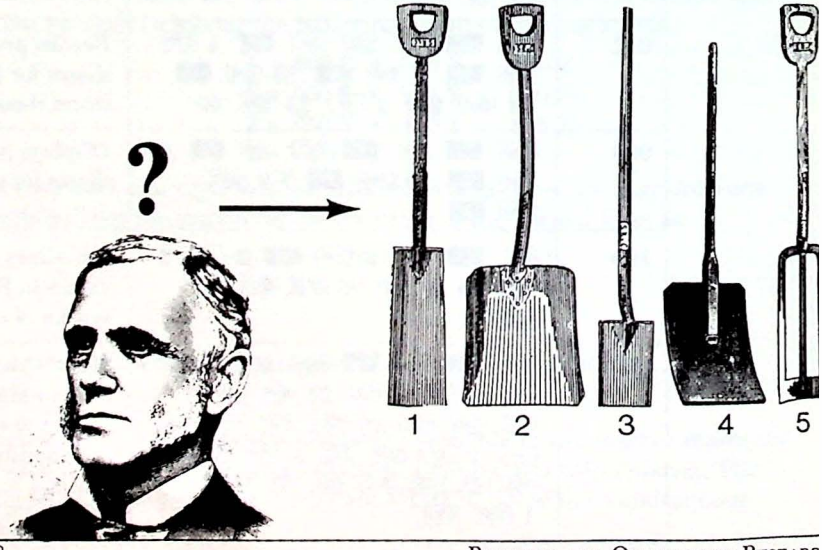
Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [2nd] [A'] 23 [STO] 00 26 [STO] 01 3 [STO] 2 [STO] 03 5 [STO] 4 [CLR] [INV] [SBR]	Initializes registers to be used in storing data and computing probable market shares.
020	[2nd] [Lbl] [A] [STO] [2nd] [Ind] 0 [2nd] [Op] 20 [R/S]	Stores market shares.
027	[2nd] [Lbl] [B] [STO] [2nd] [Ind] 1 [2nd] [Op] 21 [R/S]	Stores transition matrix.
034 040 057	[2nd] [Lbl] [C] [2nd] [A'] [CLR] [STO] [2nd] [Ind] 04 [2nd] [Lbl] [2nd] [C'] [RCL] [2nd] [Ind] 0 [×] [RCL] [2nd] [Ind] 1 [=] [SUM] [2nd] [Ind] 4 [2nd] [Op] 20 [2nd] [Op] 21 [2nd] [Dsz] 2 [2nd] [C'] 23 [STO] 00 3 [STO] 2 [2nd] [Op] 24 [CLR] [STO] [2nd] [Ind] 4 [2nd] [Dsz] 3 [2nd] [C'] 3 [STO] 02 5 [STO] 04 23 [STO] 00	Computes probable market shares for the next period. Stores them in R_{05}–R_{07}.
082	[2nd] [Lbl] [D] [RCL] [2nd] [Ind] 4 [STO] [2nd] [Ind] 0 [2nd] [Op] 20 [2nd] [Op] 24 [2nd] [Dsz] 2 [D] 23 [STO] 00	Recalls probable market shares for this period and stores them in R_{23} - R_{25}.
099	[2nd] [Lbl] [2nd] [D'] [RCL] [2nd] [Ind] 0 [2nd] [Op] 20 [2nd] [Prt] [R/S] [GTO] [2nd] [D']	Displays probable market shares for next period.
109	[2nd] [Lbl] [E] 3 [2nd] [Pgm] 2 [A] 8 [STO] 01 1 [STO] 20 [CLR] [STO] 21 [STO] 22	Initializes ML-02. Stores b_i values in R_{20} - R_{22} for solving system of equations.
126	[2nd] [Lbl] [2nd] [E'] [RCL] 26 [−] 1 [=] [SBR] [STO] [RCL] 29 [SBR] [STO] 1 [SBR] [STO] [RCL] 27 [SBR] [STO] [RCL] 30 [−] 1 [=] [SBR] [STO] 1 [SBR] [STO] [RCL] 28 [SBR] [STO] [RCL] 31 [SBR] [STO] 1 [SBR] [STO]	Stores values of matrix to be used in ML-02 to solve the system of equations to find the equilibrium market shares.

Step #	Program Keystrokes	Purpose/Comments
167	CLR 2nd Pgm 2 C x≥t 0 2nd x≥t × CLR 2nd Pgm 2 E 20 STO 00	Calls ML-02 to solve the system of equations. If the determinant equals 0, no solution exits and a flashing display will result.
183	2nd Lbl 2nd B' RCL 2nd Ind 0 2nd Op 20 2nd Prt R/S GTO 2nd B'	Displays equilibrium market share values.
193	2nd Lbl STO STO 2nd Ind 1 2nd Op 21 INV SBR	Subroutine used to store values of matrix to be used in ML-02.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}	Used
R_{01} - R_{11}	ML-02
R_{23} - R_{25}	Market Shares
R_{26} - R_{34}	Transition Matrix



Using the Program

Procedure	Press	Display/Comments
Fix decimal for desired accuracy.	2nd Fix n	
Initialize.	2nd A	0.
Enter current market shares.		
MS_1	A	MS_1
MS_2	A	MS_2
MS_3	A	MS_3
Enter transition matrix by columns		
T_{11}	B	T_{11}
T_{21}	B	T_{21}
T_{31}	B	T_{31}
. . .	. . .	. . .
T_{33}	B	T_{33}
Calculate next period's probable market shares.	C	MS_1*
	R/S	MS_2*
	R/S	MS_3*
To calculate another future period's market shares, repeat these instructions as many times as needed.	C	MS_1*
	R/S	MS_2*
	R/S	MS_3*
Calculate equilibrium market shares.	E	EMS_1*
	R/S	EMS_2*
	R/S	EMS_3*
Note: In this calculation some intermediate values are computed by ML-02 and printed. If you are using a PC-100A, be sure to wait until the entire calculation is completed before continuing.		

*This value is printed if the PC-100A is used.

6

Markov Analysis

Sample Problem

Use the transition matrix and April market shares given in the *Facts/Theory* section to compute the equilibrium shares. The matrix and market shares are shown again here for convenience.

$$\text{Transition matrix} = \begin{pmatrix} .900 & .033 & .067 \\ .060 & .910 & .030 \\ .050 & .125 & .825 \end{pmatrix}$$

$$\text{April 1 market shares} = \begin{pmatrix} .31 & .49 & .20 \end{pmatrix}$$

Procedure	Press	Display/Comments
Enter program.		0
Fix decimal for desired accuracy. (We'll use 4.)	2nd fix 4	0.0000
Initialize.	2nd A	0.0000
Enter April market shares.		
MS_1	.31 A	0.3100
MS_2	.49 A	0.4900
MS_3	.20 A	0.2000
Enter transition matrix.		
T_{11}	.900 B	0.9000
T_{21}	.060 B	0.0600
T_{31}	.050 B	0.0500
T_{12}	.033 B	0.0330
T_{22}	.910 B	0.9100
T_{32}	.125 B	0.1250
T_{13}	.067 B	0.0670
T_{23}	.030 B	0.0300
T_{33}	.825 B	0.8250
Calculate May 1 market shares.	C	0.3184
	R/S	0.4811
	R/S	0.2005

Procedure	Press	Display/Comments
Calculate June 1 market shares.	C R/S R/S	0.3255 0.4734 0.2012
Calculate equilibrium market shares.	E R/S R/S	0.3618 0.4266 0.2116
Note: In this calculation some intermediate values are computed by ML-02 and printed. If you are using a PC-100A, be sure to wait until the entire calculation is completed before continuing.		

FURTHER EXPLORATIONS

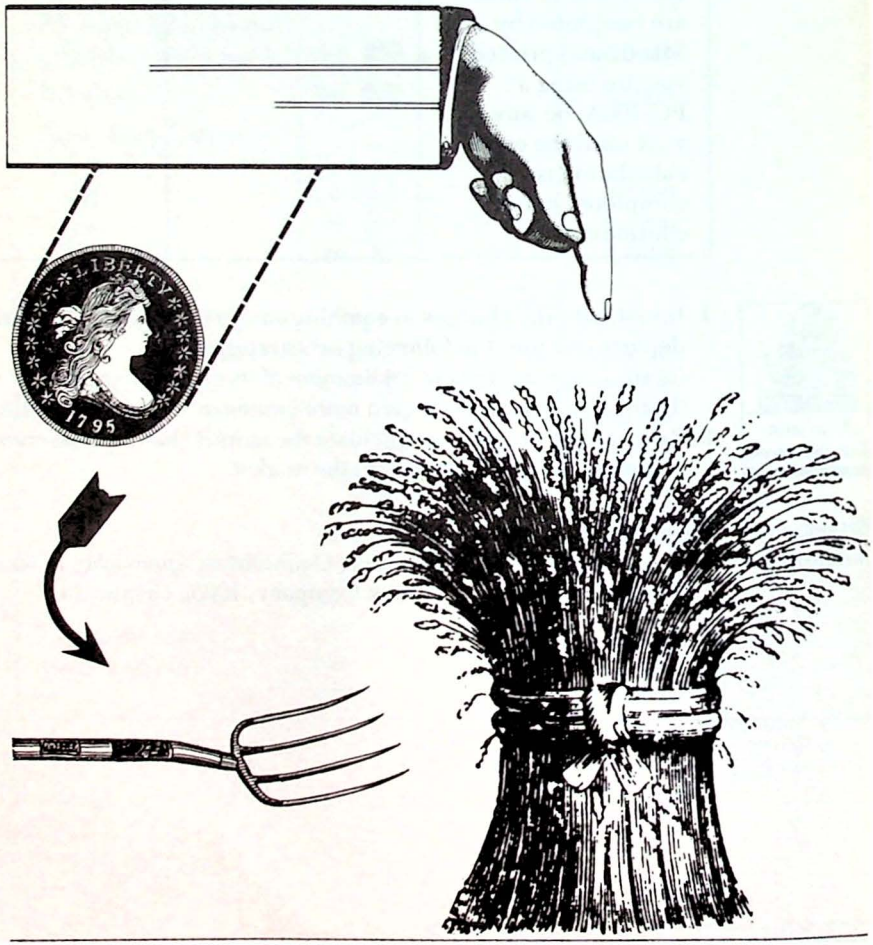
1. Investigate the changes in equilibrium market shares if the marketing department uses the following two strategies:
   (a) the company tries to retain more of its own customers.
   (b) the company tries to gain more customers from its competitors.
2. Rewrite the program to calculate the market shares for different numbers of companies (other than 3) in the market.

REFERENCES/ INFORMATION

Master Library, pp. 9-10.

R. I. Levin and C. A. Kirkpatrick, *Quantitative Approaches to Management* (New York: McGraw-Hill Book Company, 1965), chapter 14.

7 Economics



Introduction

Certain analytical and empirical concepts are important in modern economics, and the study of these concepts is usually helped by examples. In the case of empirical concepts (those dealing with specific measures of economic activities, such as price indexes, unemployment rates, seasonal adjustment, rate of growth, cyclical movements, etc.), an example representing a reasonably realistic situation can help you to understand the advantages and shortcomings of these concepts when you are evaluating policies or testing theories.

In analyzing the application of economic concepts to real-life situations, a numerical example may well help you to understand and follow the logic of the theory and its application. For instance, the law of variable proportions in studies of production and all the rules about short-run cost functions that follow from the law are typical of a microeconomic theory in which numerical examples point out the utility of the concept and increase your understanding of its logic.

In both these cases you can profit considerably from the use of a programmable calculator. These calculations have traditionally been done either by using a slide-rule calculator or a computer, but both of these methods have disadvantages. The slide-rule calculator does not allow for realistic examples because these usually require long, repetitive calculations. With a computer "real life" numbers can be used and realistic results obtained; however, the whole purpose of the exercise may be lost because it is difficult to follow individual steps or at least individual groups of steps in the calculations well enough to develop full understanding of the concept. Also, involvement in traditional computer programming and card punching requires quite a bit of time. On the other hand, if ready-made programs are used, you are presented with final results only, thus losing the opportunity to see the concept in action and understand its logic and function.

The advantages of a programmable calculator are its accessibility and versatility. It is easy to follow the calculations step by step and see results as they are obtained, and repetitive calculations are simplified by the ability to enter a program and run it as many times as necessary, changing variable data as needed.

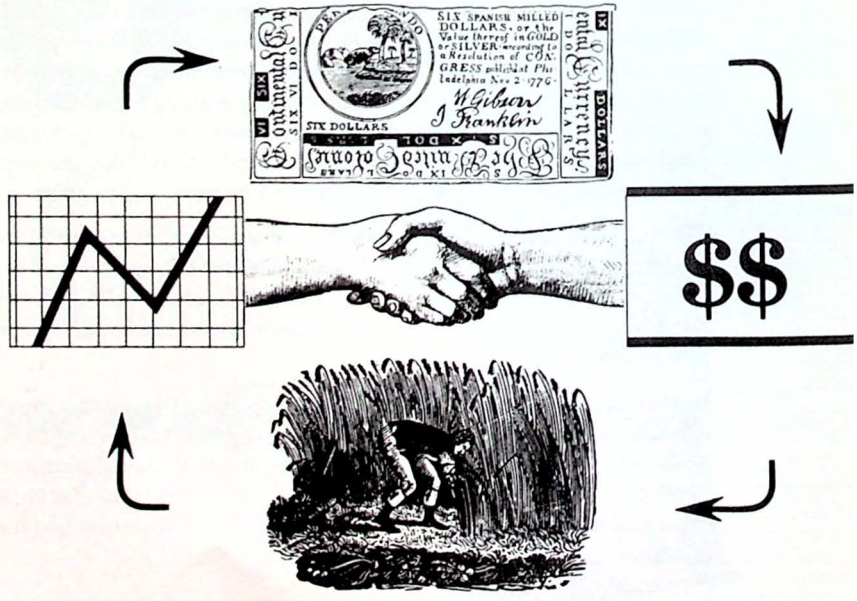
In this chapter we have three examples showing the use of a programmable calculator to improve the understanding of empirical concepts. One deals with calculating a seasonal index, the second with calculating a reference cycle for the latest business cycle, and the third with calculating fluctuation around a trend line. In all of these examples the advantage of avoiding long, repetitive arithmetic calculations is clear.

7

Introduction

In theory courses analytical concepts are introduced and sometimes accompanied by numerical examples to stress the meaning of the implied relationships involved. The numerical examples in this case call for repetitive solutions of one or several equations, substituting consecutive numbers as independent variables. Again, this is a time-consuming task for a slide-rule calculator or an impersonal grinding of results when a computer is used. The advantage of a programmable calculator here is that the actual calculations can be done right in the classroom or in the field without much loss of time. The possible effects of changes in the parameters of the respective equations can also be studied without rewriting the program or queuing up for the use of the computer.

Two examples of this type are also included in the chapter. One is a numerical example to explore the effects of the interaction of the investment multiplier and the accelerator on the cyclical growth of the economy. The other is an example of a short-run production function and the various cost functions that follow from it. In both cases actual figures are generated quickly once the constants of the equations have been decided on, and you can easily follow the direction and magnitude of change in the dependent variables.


SIX SPANISH MILLED DOLLARS, or the Value thereof in GOLD or SILVER, according to a Resolution of CONGRESS, published at Philadelphia Nov. 2. 1776.
W. Gibson
J. Franklin
SIX DOLLARS
$$

Seasonal Index (Ratio to Centered Average)

OBJECTIVE

Most monthly economic data, when published, are "seasonally adjusted." To understand better the meaning of the seasonal adjustment and its precision as a measure of the effect of the "season" on particular time series, one should attempt to do an actual calculation of a seasonal index. Such calculation consists essentially of two steps. The first step involves a large number of repetitive operations, while the second step implies certain value judgments depending on the results of the first step. Because of the time it usually requires to finish the first step, a seasonal index is seldom calculated in the classroom, but a programmable calculator allows for relatively fast completion of the first step and thus can give you actual experience in what kinds of value judgments are required in the second step.

FACTS/ THEORY

The first step in the construction of a seasonal index is to calculate the ratios of actual values to centered averages for each month of a time series. We do this by first finding the weighted 13 months' average centered at a given month. Algebraically, this procedure may be expressed as

$$\bar{x}_1 = \frac{x_{-6} + 2(x_{-5} + x_{-4} + x_{-3} + x_{-2} + x_{-1} + x_1 + x_2 + x_3 + x_4 + x_5 + x_6) + x_7}{24}$$

where $\bar{x}_1$ is the weighted centered average for month x_1 (x_{-1} to x_{-6} are the six months preceding month x_1, and x_2 to x_7 are the six months following month x_1).

Then we divide the actual value for x_1 by its centered average to find the ratio for month x_1, multiplying the result by 100 to convert the ratio into standard form:

$$r = \frac{x_1}{\bar{x}_1} \times 100$$

For one year (twelve months) we need 12 ratios of actual values to the centered average. This ratio is supposed to approximate the relative effect of the season. To improve the measure of this effect by eliminating irregular as well as possible cyclical effects that are present in the ratio, we need more than one year's experience. Probably ten years, meaning altogether 120 monthly ratios, should be used in the calculation. This implies 120 centered weighted averages and their ratio to actual values — clearly a long, tedious calculation.

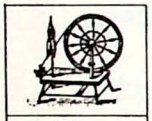
The second step in the calculation of a seasonal index requires the best utilization of the information obtained in the first step. This could be done by discovering a possible trend in the seasonal effect over time, identification of extreme values of the ratios due to specific irregular effects such as strikes, etc. Finally, the decision has to be made as to how one seasonal index (ratio × 100) is selected. Usually it will be an average of all or selected individual ratios, etc.

7

Seasonal Index (Ratio to Centered Average)

The following program allows you to input your original data and calculate the monthly ratios. The first segment of the program evaluates the above formulas, based on 13 months' data stored in memory registers 1 through 13. The second portion automatically advances the stored data by one memory register, making it necessary for you to enter only one new month's data each time you run the program.

The program will store a maximum of 15 calculated results in the TI Programmable 58 and a maximum of 45 results in the TI Programmable 59 without repartitioning. If you are calculating ratios to centered averages beyond these levels, remember to repartition the data storage and program storage registers appropriately before entering the program. (See pages V-22, 29, and 42 in *Personal Programming* for a discussion of repartitioning.)



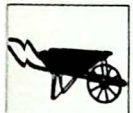
The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A RCL 1 + (RCL 2 + RCL 3 + RCL 4 + RCL 5 + RCL 6 + RCL 7 + RCL 8 + RCL 9 + RCL 10 + RCL 11 + RCL 12) × 2 + RCL 13 = ÷ 24 = 1/x × RCL 7 × 100 = STO 2nd Ind 14 2nd Prt 2nd Adv R/S	Calculates monthly ratio to centered average. *Note:* The STO 2nd Ind 14 key sequence commands the program to store the generated data in the memory location contained in memory 14.
063	2nd Exc 13 2nd Exc 12 2nd Exc 11 2nd Exc 10 2nd Exc 9 2nd Exc 8 2nd Exc 7 2nd Exc 6 2nd Exc 5 2nd Exc 4 2nd Exc 3 2nd Exc 2 2nd Exc 01 1 SUM 14 A	Stores new monthly data in memory 13 and rolls previously stored data forward one memory location. Then sums 1 to memory 14 for next indirect storage of generated ratio.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

Seasonal Index (Ratio to Centered Average)

Register Contents					
R_{00}		R_{10}	Used	R_{20}	(remaining
R_{01}	Used	R_{11}	Used	R_{21}	memories
R_{02}	Used	R_{12}	Used	R_{22}	used as
R_{03}	Used	R_{13}	Used	R_{23}	needed
R_{04}	Used	R_{14}	Used	R_{24}	for storage
R_{05}	Used	R_{15}	r_1	R_{25}	of generated
R_{06}	Used	R_{16}	r_2	R_{26}	ratios)
R_{07}	Used	R_{17}	r_3	R_{27}	
R_{08}	Used	R_{18}	r_4	R_{28}	
R_{09}	Used	R_{19}	r_5	R_{29}	



Using the Program

Procedure	Press	Display/Comments
Clear memories and set decimal place for desired accuracy.	2nd CMs	0.
	2nd fix 2	0.00
Enter data for first 13 months in memory registers 1 through 13.	data STO 01	data x_{-6}
	data STO 02	data x_{-5}
	data STO 03	data x_{-4}
	data STO 04	data x_{-3}
	data STO 05	data x_{-2}
	data STO 06	data x_{-1}
	data STO 07	data x_1
	data STO 08	data x_2
	data STO 09	data x_3
	data STO 10	data x_4
	data STO 11	data x_5
	data STO 12	data x_6
	data STO 13	data x_7
Enter 15 in memory register 14.	15 STO 14	15. First generated ratio will be stored in memory register 15; second in memory 16, third in memory 17, etc.
Calculate ratio to centered average for first month.	A	$ratio_1$ ° (Stored in memory 15.)

7

Seasonal Index (Ratio to Centered Average)

Procedure	Press	Display/Comments
Enter 14th month's data, roll data forward, and calculate next ratio.	data R/S	$ratio_2$° (Stored in memory 16.)
Enter 15th month's data, roll data forward, and calculate next ratio.	data R/S	$ratio_3$° (Stored in memory 17.)
Continue above procedure until all ratios have been calculated.		

°This value is printed if the PC-100A is used.

The monthly ratios may be recalled from memories as needed, beginning with register 15.

Sample Problem

Using the following monthly production data, calculate ratios to centered averages for January-December, 1976.°

July '75	114.5	July '76	126.0
Aug. '75	121.4	Aug. '76	131.7
Sept. '75	125.9	Sept. '76	134.6
Oct. '75	125.4	Oct. '76	134.0
Nov. '75	123.8	Nov. '76	132.2
Dec. '75	119.8	Dec. '76	128.1
Jan. '76	122.1	Jan. '77	128.4
Feb. '76	127.9	Feb. '77	133.6
Mar. '76	128.6	Mar. '77	135.9
Apr. '76	128.7	Apr. '77	136.2
May '76	129.9	May '77	137.4
June '76	133.5	June '77	141.8

°Unadjusted Index of Industrial Production for 1975 and 1976 is from *Survey of Current Business,* U.S. Department of Commerce, Washington, D.C., Volumes 56 (August 1976) and 57 (August 1977) respectively, p. S-4.

Procedure	Press	Display/Comments
Enter program.		0
Clear memories and fix decimal at 2 places.	2nd CMS 2nd Fix 2	0. 0.00
Enter data for first 13 months.		
(July 1975)	114.5 STO 01	114.50 x_{-6}
(Aug. 1975)	121.4 STO 02	121.40 x_{-5}
(Sept. 1975)	125.9 STO 03	125.90 x_{-4}
(Oct. 1975)	125.4 STO 04	125.40 x_{-3}
(Nov. 1975)	123.8 STO 05	123.80 x_{-2}
(Dec. 1975)	119.8 STO 06	119.80 x_{-1}
(Jan. 1976)	122.1 STO 07	122.10 x_1
(Feb. 1976)	127.9 STO 08	127.90 x_2
(Mar. 1976)	128.6 STO 09	128.60 x_3
(Apr. 1976)	128.7 STO 10	128.70 x_4
(May 1976)	129.9 STO 11	129.90 x_5
(June 1976)	133.5 STO 12	133.50 x_6
(July 1976)	126 STO 13	126.00 x_7
Enter 15 in memory 14.	15 STO 14	15.00
Calculate ratio for Jan. 1976.	A	97.21
Enter next month's data (Aug. 1976), and calculate ratio for Feb. 1976.	131.7 R/S	101.10
Enter next monthly data (Sept. 1976), and calculate ratio for March, 1976.	134.6 R/S	101.02

Seasonal Index (Ratio to Centered Average)

Repeat the above procedure until ratios have been calculated for twelve months (January through December 1976). The following results will be generated.

Month (1976)	*Ratio to Centered Average*	*Memory Register*
January	97.21	15
February	101.10	16
March	101.02	17
April	100.53	18
May	100.91	19
June	103.15	20
July	96.90	21
August	100.89	22
September	102.69	23
October	101.75	24
November	99.91	25
December	96.33	26

Further Explorations

1. Calculate seasonal indexes, using the monthly ratios to centered averages over a period of several years. Each August issue of the *Survey of Current Business*, published monthly by the U.S. Department of Commerce, provides data on unadjusted indexes of industrial production for the previous twelve months (July to June). These averages can then form a basis for selection of an actual seasonal index. The selection procedure may imply a test for trend in the ratios for individual months, the calculation of simple averages, or the calculation of modified averages.

2. Prepare a program to calculate seasonal indexes based on quarterly data, rather than monthly data.

References/ Information

Seasonal indexes are discussed in many textooks. A detailed explanation of the calculation and use of seasonal indexes is presented in Frederick E. Croxton and Dudley J. Cowden, *Applied General Statistics*, 2nd ed. (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1960), pp. 320-364.

Reference Cycle

OBJECTIVE

Studying the behavior of different time series during business cycles may be useful for forecasting, evaluating current situations, or simply understanding economic relationships better. For comparisons of the behavior of various time series during the same business cycle or the same time series during different business cycles, data should be converted to a standard form which describes each time series in terms of nine reference values.

The calculations necessary to convert data into the standard form can be very time consuming. However, when you need information about how several time series behaved during the same cycle in general economic activity, your programmable calculator can help to reduce calculation time.

FACTS/ THEORY

Each of the nine "points on the reference cycle" is a partial average relative to the arithmetic mean value ($\bar{x}$) of the whole cycle. The first point is the mean of three observations — the month preceding the trough, the trough, and the month following the trough — divided by the mean value of all observations from the trough to the next trough and multiplied by 100.

$$\text{Point } 1 = \frac{\sum_{0}^{2} x_i}{3\bar{x}} \times 100$$

The second, third, and fourth points are averages of the first, second, and third thirds of the expansion phase of the cycle. (The algebraic notation used here refers to the reference cycle 1973–1975 only. For other cycles, the number of observations in each third of the expansion and contraction phases would be different.)

$$\text{Point } 2 = \frac{\sum_{1}^{12} x_i}{12\bar{x}} \times 100$$

$$\text{Point } 3 = \frac{\sum_{13}^{24} x_i}{12\bar{x}} \times 100$$

$$\text{Point } 4 = \frac{\sum_{25}^{36} x_i}{12\bar{x}} \times 100$$

The fifth point is the three-month centered average of the peak month, and the next three points are again averages representing the first, second, and third thirds of the contraction phase of the cycle.

$$\text{Point } 5 = \frac{\sum_{36}^{38} x_i}{3\bar{x}} \times 100$$

$$\text{Point } 6 = \frac{\sum_{37}^{42} x_i}{6\bar{x}} \times 100$$

$$\text{Point } 7 = \frac{\sum_{43}^{47} x_i}{5\bar{x}} \times 100$$

$$\text{Point } 8 = \frac{\sum_{48}^{53} x_i}{6\bar{x}} \times 100$$

Finally, the ninth point is the centered average of the second trough of this particular reference cycle.

$$\text{Point } 9 = \frac{\sum_{52}^{54} x_i}{3\bar{x}} \times 100$$

The arithmetic mean of the time series for the whole reference cycle is expressed as:

$$\bar{x} = \frac{\sum_{1}^{53} x_i}{53}$$

The program included here can be used to calculate the standard form of the last full business cycle (November 1973-November 1975) for any time series available monthly, in seasonally adjusted values. Our example problem calculates the nine points on the reference cycle for the Index of Industrial Production. For notation, $x_0, x_1, x_2 \ldots x_{54}$ are used to represent the monthly values of the time series from October 1970 to April 1975.

THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl Inx 1 SUM 0 RCL 0 INV SBR	Counting subroutine for x_i entries.
008	2nd Lbl A SUM 1 SUM 2 SBR Inx R/S	The next 9 subroutines are summation routines for entry of x_1 through x_{53}.
017	2nd Lbl B SUM 2 SBR Inx R/S	
024	2nd Lbl C SUM 3 SBR Inx R/S	
031	2nd Lbl D SUM 4 SBR Inx R/S SUM 4 SUM 5 SBR Inx R/S	
045	2nd Lbl E SUM 5 SUM 6 SBR Inx R/S	
054	2nd Lbl 2nd A' SUM 6 SBR Inx R/S	
061	2nd Lbl 2nd B' SUM 7 SBR Inx R/S	
068	2nd Lbl 2nd C' SUM 8 SBR Inx R/S	
075	2nd Lbl 2nd D' SUM 8 SUM 9 SBR Inx R/S	
084	2nd Lbl 2nd E' RCL 2 + RCL 3 + RCL 4 + RCL 6 + RCL 7 + RCL 8 = ÷ 53 = STO 10	Calculates arithmetic mean.
110	RCL 1 SBR 1/x STO 11 R/S	Calculates point 1.
117	RCL 2 SBR x^2 STO 12 R/S	Calculates point 2.
124	RCL 3 SBR x^2 STO 13 R/S	Calculates point 3.
131	RCL 4 SBR x^2 STO 14 R/S	Calculates point 4.

Reference Cycle

Step #	Program Keystrokes	Purpose/Comments
138	RCL 5 SBR 1/x STO 15 R/S	Calculates point 5.
145	RCL 6 SBR √x STO 16 R/S	Calculates point 6.
152	RCL 7 ÷ (5 × RCL 10) × 100 = 2nd Prt STO 17 R/S	Calculates point 7.
170	RCL 8 SBR √x STO 18 R/S	Calculates point 8.
177	RCL 9 SBR 1/x STO 19 R/S	Calculates point 9.
184	2nd Lbl 1/x ÷ (3 × RCL 10) × 100 = 2nd Prt INV SBR	The remaining subroutines are used within subroutine E' to provide proper divisor for calculation of each point.
200	2nd Lbl √x ÷ (6 × RCL 10) × 100 = 2nd Prt INV SBR	
216	2nd Lbl x² ÷ (12 × RCL 10) × 100 = 2nd Prt INV SBR	
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

Register Contents

R_{00}	Entry counter	R_{05}	Used	R_{10}	$\bar{x}$	R_{15}	Point 5
R_{01}	Used	R_{06}	Used	R_{11}	Point 1	R_{16}	Point 6
R_{02}	Used	R_{07}	Used	R_{12}	Point 2	R_{17}	Point 7
R_{03}	Used	R_{08}	Used	R_{13}	Point 3	R_{18}	Point 8
R_{04}	Used	R_{09}	Used	R_{14}	Point 4	R_{19}	Point 9

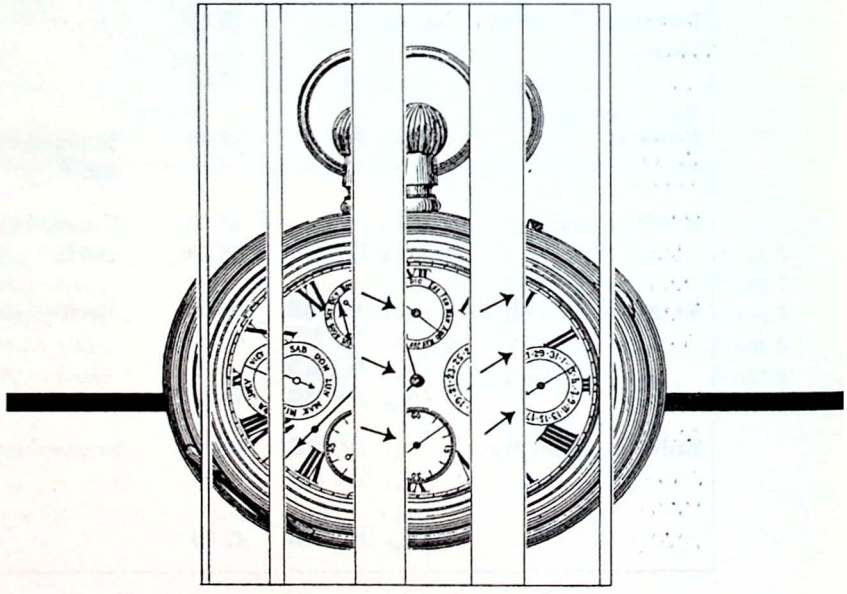
USING THE PROGRAM

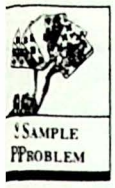
Procedure	Press	Display/Comments	
Clear memories and fix decimal for desired display format.	2nd CMs 2nd Fix n		We'll use two-place decimal setting for convenience.
Enter monthly data x_0 and x_{54}.	x_0 STO 01 x_{54} STO 09	x_0 x_{54}	These data are not included in finding $\bar{x}$.
Enter x_1 and x_2.	x_1 A x_2 A	1.00 2.00	Summed into memories 1 and 2. Note that the program counts the x_i entries.
Enter x_3 through x_{12}, pressing B after each entry.	x_3 B x_4 B . . . x_{12} B	3.00 4.00 . . . 12.00	Summed into memory 2.
Enter x_{13} through x_{24}, pressing C after each entry.	x_{13} C x_{14} C . . . x_{24} C	13.00 14.00 . . . 24.00	Summed into memory 3.
Enter x_{25} through x_{35}, pressing D after each entry.	x_{25} D x_{26} D . . . x_{35} D	25.00 26.00 . . . 35.00	Summed into memory 4.
Enter x_{36}.	x_{36} R/S	36.00	Summed into memories 4 and 5.
Enter x_{37} and x_{38}.	x_{37} E x_{38} E	37.00 38.00	Summed into memories 5 and 6.
Enter x_{39} through x_{42}.	x_{39} 2nd A' x_{40} 2nd A' x_{41} 2nd A' x_{42} 2nd A'	39.00 40.00 41.00 42.00	Summed into memory 6.
Enter x_{43} through x_{47}.	x_{43} 2nd B' x_{44} 2nd B' . . . x_{47} 2nd B'	43.00 44.00 . . . 47.00	Summed into memory 7.

Reference Cycle

Procedure	Press	Display/Comments
Enter x_{48} through x_{51}.	x_{48} 2nd C'	48.00 Summed into memory 8.
	x_{49} 2nd C'	49.00
	x_{50} 2nd C'	50.00
	x_{51} 2nd C'	51.00
Enter x_{52} and x_{53}.	x_{52} 2nd D'	52.00 Summed into memories
	x_{53} 2nd D'	53.00 and 9.
Calculate arithmetic mean (stored in memory 10) and the nine points on the reference cycle.	2nd E'	Point 1°
	R/S	Point 2°
	R/S	Point 3°
	R/S	Point 4°
	R/S	Point 5°
	R/S	Point 6°
	R/S	Point 7°
	R/S	Point 8°
	R/S	Point 9°

°This value is printed if the PC-100A is used.





Using the following time series, calculate the nine points on the reference cycle for November 1970 through March 1975.*

Oct. '70	103.7	Apr. '72	112.8	Oct. '73	127.0
Nov. '70	102.6	May '72	113.2	Nov. '73	127.5
Dec. '70	104.6	June '72	114.4	Dec. '73	126.5
Jan. '71	105.3	July '72	115.1	Jan. '74	125.5
Feb. '71	105.7	Aug. '72	116.3	Feb. '74	124.7
Mar. '71	105.5	Sept. '72	117.6	Mar. '74	124.7
Apr. '71	106.2	Oct. '72	119.2	Apr. '74	124.9
May '71	107.4	Nov. '72	120.2	May '74	125.7
June '71	107.4	Dec. '72	121.1	June '74	125.8
July '71	106.8	Jan. '73	122.2	July '74	125.5
Aug. '71	105.6	Feb. '73	123.4	Aug. '74	125.2
Sept. '71	107.1	Mar. '73	123.7	Sept. '74	125.6
Oct. '71	106.8	Apr. '73	124.1	Oct. '74	124.8
Nov. '71	107.4	May '73	124.9	Nov. '74	121.7
Dec. '71	108.1	June '73	125.6	Dec. '74	117.4
Jan. '72	108.7	July '73	126.7	Jan. '75	113.7
Feb. '72	110.0	Aug. '73	126.4	Feb. '75	111.2
Mar. '72	111.2	Sept. '73	126.8	Mar. '75	110.0
				Apr. '75	109.9

*Seasonally adjusted Index of Industrial Production is from *Survey of Current Business*, U.S. Department of Commerce, Washington, D.C., Vols. 51-56, p. S-4 in each issue.

Procedure	Press	Display/Comments	
Enter program.		**0**	
Clear memories	2nd CMs	**0.**	
and fix decimal.	2nd Fix 2	**0.00**	
Enter data for Oct. 1970 (x_0)	**103.7** STO **01**	**103.70**	Note that x_0 and x_{54} must be entered *before* running program.
Enter data for April 1975 (x_{54}).	**109.9** STO **09**	**109.90**	
Enter data for Nov. 1970 (x_1) and Dec. 1970 (x_2).	**102.6** A **104.6** A	**1.00** **2.00**	The program counts the entries for you.

Procedure	Press	Display/Comments	
Enter data for Jan. 1971 (x_3) through Oct. 1971 (x_{12}). (Press [B] after each entry.)	105.3 [B] 105.7 [B] . . . 106.8 [B]	3.00 4.00 . . . 12.00	
Enter data for Nov. 1971 (x_{13}) through Oct. 1972 (x_{24}). (Press [C] after each entry.)	107.4 [C] 108.1 [C] . . . 119.2 [C]	13.00 14.00 . . . 24.00	
Enter data for Nov. 1972 (x_{25}) through Sept. 1973 (x_{35}).	120.2 [D] 121.1 [D] . . . 126.8 [D]	25.00 26.00 . . . 35.00	
Enter data for Oct. 1973 (x_{36}).	127 [R/S]	36.00	Note that this data point requires special entry.
Enter data for Nov. 1973 (x_{37}) and Dec. 1973 (x_{38}).	127.5 [E] 126.5 [E]	37.00 38.00	
Enter data for Jan. 1974 (x_{39}) through Apr. 1974 (x_{42}).	125.5 [2nd] [A'] . . . 124.9 [2nd] [A']	39.00 . . . 42.00	
Enter data for May 1974 (x_{43}) through Sept. 1974 (x_{47}).	125.7 [2nd] [B'] . . . 125.6 [2nd] [B']	43.00 . . . 47.00	
Enter data for Oct. 1974 (x_{48}) through Jan. 1975 (x_{51}).	124.8 [2nd] [C'] . . . 113.7 [2nd] [C']	48.00 . . . 51.00	
Enter data for Feb. 1975 (x_{52}) and Mar. 1975 (x_{53}).	111.2 [2nd] [D'] 110 [2nd] [D']	52.00 53.00	

All data have now been entered. You are ready to calculate the arithmetic mean and the nine points on the reference cycle.

Procedure	Press	Display/Comments
Calculate points:		
Point 1.	2nd Σ	88.63 Memory 11.
Point 2.	R/S	90.58 Memory 12.
Point 3.	R/S	96.49 Memory 13.
Point 4.	R/S	106.33 Memory 14.
Point 5.	R/S	108.61 Memory 15.
Point 6	R/S	107.44 Memory 16.
Point 7.	R/S	107.38 Memory 17.
Point 8.	R/S	99.60 Memory 18.
Point 9.	R/S	94.38 Memory 19.
If the arithmetic mean is needed, recall from memory.	RCL 10	116.93 Arithmetic mean.

FURTHER PLORATIONS

It may be desirable to make comparisons of the behavior of a particular time series during different cycles. For that you'll need to write a new program for each business cycle. The timing of post-WW II cycles has been identified by the National Bureau of Economic Research as follows:

Trough	*Peak*	*Trough*	*Duration (months)*
Oct. 1945	Nov. 1948	Oct. 1949	48
Oct. 1949	July 1953	May 1954	55
May 1954	Aug. 1957	Apr. 1958	47
Apr. 1958	Apr. 1960	Feb. 1961	34
Feb. 1961	Nov. 1969	Nov. 1970	117
Nov. 1970	Nov. 1973	Mar. 1975	52

FERENCES/ FORMATION

The method described here was developed by the National Bureau of Economic Research and is discussed in detail in various texts dealing with business cycles, specifically in R. A. Gordon, *Business Fluctuations* (New York: Harper & Brothers Publishers, 1961), pp. 265 ff.

7 Fluctuation Around the Trend Line

OBJECTIVE

In addition to the Reference Cycle method discussed earlier, another way to identify and measure the cyclical movement of a time series is to calculate the percentage differences of actual values from values projected along a trend line. This method is preferable to the Reference Cycle analysis when you are studying longer time periods and looking for different cycles superimposed on each other. The following program calculates both the trend line and the percentage deviations.

FACTS/ THEORY

It is usually assumed that any economic time series is an aggregate of seasonal, cyclical, trend, and irregular movements. To study the movements it is useful to separate the time series into the individual components. Usually the seasonal effect is identified and subtracted first. Then the trend is calculated and subtracted. The values that remain are the cyclical and irregular effects on the time series. The irregular effects may be then "averaged" out to obtain the cyclical effect as a residual.

Certain assumptions must be made about the type of trend present in the time series. For economic aggregates such as GNP, Employment, Consumption, Investment, etc., it is common to assume a trend described by a constant percentage growth over time — specifically, a straight line fitted to logarithmic values of the aggregate. The estimating technique used is usually the least square method applied to time as the independent variable and the values of the aggregate as the dependent variable.

Once the trend line has been calculated, the fluctuation around the trend is commonly measured as the percentage deviation of an actual value from its trend value. This is expressed as

$$\left(\frac{V_t}{V_{tc}} - 1\right) \times 100$$

where V_t = the actual value of the aggregate in time t, and
V_{tc} = the trend value of the value in time t.

The first part of the following program is a data-loading subroutine for calculation of the trend line. The second part calculates the percentage deviation of the actual value from the trend value.

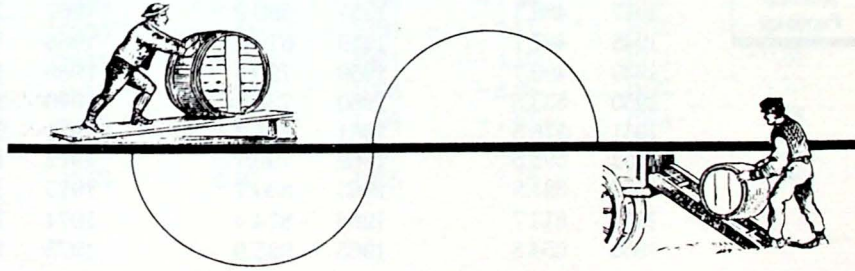
THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 7 2nd Pgm 1 SBR CLR RCL 7 x≥t R/S	Stores year in memory 7 and initializes linear regression routine.
012	2nd Lbl B Inx 2nd Σ+ R/S	Enters data.
017	2nd Lbl C STO 8 RCL 7 2nd Op 14 INV Inx 1/x × RCL 8 − 1 = × 100 = 2nd Op 27 2nd Prt R/S	Calculates percentage deviation.
	LRN RST	Exits learn mode.

Note: Print command may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}		R_{05}	Used
R_{01}	Used	R_{06}	Used
R_{02}	Used	R_{07}	Year
R_{03}	Used	R_{08}	V_t
R_{04}	Used	R_{09}	



7

Fluctuation Around the Trend Line

Using the Program

Procedure	Press	Display/Comments
Select decimal setting for desired accuracy.	2nd fix n	We'll use two-place decimal for convenience.
Enter first year (t) for both subroutines and initialize.	year A	0.00
Enter aggregate values.		
1st year	V_1 B	1.00
2nd year	V_2 B	2.00
3rd year	V_3 B	3.00
. . .	. . .	. . .
nth year	V_n B	n.00
Calculate percentage deviations.		
1st year	V_1 C	V_{1c} *
2nd year	V_2 C	V_{2c} *
3rd year	V_3 C	V_{3c} *
. . .	. . .	. . .
nth year	V_n C	V_{nc} *

*This value is printed if the PC-100A is used.

Sample Problem

Calculate the percentage deviations from a trend line of Gross National Product in constant prices for 1946 to 1975*:

Year	*GNP*	*Year*	*GNP*	*Year*	*GNP*
1946	477.0	1956	668.8	1966	981.0
1947	468.3	1957	680.9	1967	1007.7
1948	487.7	1958	679.5	1968	1051.8
1949	490.7	1959	720.4	1969	1078.8
1950	533.5	1960	736.8	1970	1075.3
1951	576.5	1961	755.3	1971	1107.5
1952	598.5	1962	799.1	1972	1171.1
1953	621.8	1963	830.7	1973	1235.0
1954	613.7	1964	874.4	1974	1214.0
1955	654.8	1965	925.9	1975	1191.7

*These figures are from *Economic Report of the President 1977*, United States Government Printing Office, Washington, D.C., 1977, p. 188.

Procedure	Press	Display/Comments
Enter program.		0
Set decimal at two places.	2nd fix 2	0.00
Enter first year for both subroutines.	1946 A	0.00
Enter data.		
1st year	477 B	1.00
2nd year	468.3 B	2.00
3rd year	487.7 B	3.00
. . .	. . .	. . .
30th year	1191.7 B	30.00
Calculate percentage deviation.		
1st year	477 C	2.44
2nd year	468.3 C	−2.89
3rd year	487.7 C	−2.35
. . .	. . .	. . .
30th year	1191.7 C	−7.36

The full results are:

Year	*% Deviation*	*Year*	*% Deviation*	*Year*	*% Deviation*
1946	2.44	1956	1.18	1966	4.54
1947	−2.89	1957	−0.54	1967	3.68
1948	−2.35	1958	−4.16	1968	4.50
1949	−5.13	1959	−1.89	1969	3.49
1950	−0.41	1960	−3.11	1970	−0.40
1951	3.91	1961	−4.10	1971	−0.95
1952	4.16	1962	−2.03	1972	1.13
1953	4.49	1963	−1.67	1973	2.98
1954	−0.42	1964	−0.06	1974	−2.26
1955	2.59	1965	2.18	1975	−7.36

Fluctuation Around the Trend Line

Further Explorations

1. When the calculation of the percentage deviation from the trend is completed, a moving average should be obtained to see the possible existence of major and minor cycles. The moving average calculation is described in *Master Library*, p. 58.

2. Identify cyclical movement around the trend for the Gross National Product for the last 88 years. The data are available in *Long Term Economic Growth 1860–1965*, U.S. Department of Commerce, Washington, D.C., 1966, pp. 166 and 167, for the years 1889 to 1953, and in *Economic Report of the President 1978*, United States Government Printing Office, Washington D.C., 1978, p. 258 for 1953 to 1977.

References/ Information

Master Library, pp. 6-8.
Personal Programming, pp. V-39,40.
See any textbook dealing with business cycles, such as: R. A. Gordon, *Business Fluctuations* (New York: Harper & Brothers Publishers, 1961), pp. 253-257.

Interaction of Multiplier and Accelerator

OBJECTIVE

The accelerator model of business cycles is discussed today in every textbook on macroeconomic theory. The original article introducing the mathematical presentation of the idea is by P. Samuelson, "Interactions Between the Multiplier Analysis and the Principle of Acceleration," *The Review of Economic Statistics*, No. 2, May 1939. Your calculator and the following program can be used to demonstrate the sensitivity of the model to changes in the parameters in the equation or to different levels of the initial disturbances.

FACTS/ THEORY

The theory is that the combined effects of the Keynesian multiplier and the investment accelerator may lead to cyclical growth in economic activity.

The underlying assumptions of the theory are that consumers' expenditures are a constant function of income, while the business investment is influenced by growth of spending by consumers. It means that if there is an increase in autonomous expenditures in the economy, coming either from the government or from the business sector, personal incomes will increase, leading to an increase in consumer spending. This, of course, requires larger production of consumer goods. The business sector will try to adjust its capital stock to meet the larger demand. That means an increase in investment. But higher investment expenditures will imply higher incomes of those who produce the capital goods and also higher spending by them on new consumer goods. This process then may lead to explosive growth.

There is an upper limit, however, on total output in the short run which ultimately prevents the production from meeting the new demand. Retardation of growth, then, implies a decline in business investment, since less additional capital is needed. Lower investment means lower incomes for producers of investment goods and, consequently, lower spending on consumer goods, and thus the process of contraction starts. Again, this process will finally reverse itself because there is a lower limit on disinvestment, that is, the downwards adjustment of the capital stock. This limit is given by the value of capital depreciation.

7

Fluctuation Around the Trend Line

Further Explorations

1. When the calculation of the percentage deviation from the trend is completed, a moving average should be obtained to see the possible existence of major and minor cycles. The moving average calculation is described in *Master Library*, p. 58.

2. Identify cyclical movement around the trend for the Gross National Product for the last 88 years. The data are available in *Long Term Economic Growth 1860–1965*, U.S. Department of Commerce, Washington, D.C., 1966, pp. 166 and 167, for the years 1889 to 1953, and in *Economic Report of the President 1978*, United States Government Printing Office, Washington D.C., 1978, p. 258 for 1953 to 1977.

References/ Information

Master Library, pp. 6-8.
Personal Programming, pp. V-39,40.
See any textbook dealing with business cycles, such as: R. A. Gordon, *Business Fluctuations* (New York: Harper & Brothers Publishers, 1961), pp. 253-257.

Interaction of Multiplier and Accelerator

OBJECTIVE

The accelerator model of business cycles is discussed today in every textbook on macroeconomic theory. The original article introducing the mathematical presentation of the idea is by P. Samuelson, "Interactions Between the Multiplier Analysis and the Principle of Acceleration," *The Review of Economic Statistics*, No. 2, May 1939. Your calculator and the following program can be used to demonstrate the sensitivity of the model to changes in the parameters in the equation or to different levels of the initial disturbances.

FACTS/ THEORY

The theory is that the combined effects of the Keynesian multiplier and the investment accelerator may lead to cyclical growth in economic activity.

The underlying assumptions of the theory are that consumers' expenditures are a constant function of income, while the business investment is influenced by growth of spending by consumers. It means that if there is an increase in autonomous expenditures in the economy, coming either from the government or from the business sector, personal incomes will increase, leading to an increase in consumer spending. This, of course, requires larger production of consumer goods. The business sector will try to adjust its capital stock to meet the larger demand. That means an increase in investment. But higher investment expenditures will imply higher incomes of those who produce the capital goods and also higher spending by them on new consumer goods. This process then may lead to explosive growth.

There is an upper limit, however, on total output in the short run which ultimately prevents the production from meeting the new demand. Retardation of growth, then, implies a decline in business investment, since less additional capital is needed. Lower investment means lower incomes for producers of investment goods and, consequently, lower spending on consumer goods, and thus the process of contraction starts. Again, this process will finally reverse itself because there is a lower limit on disinvestment, that is, the downwards adjustment of the capital stock. This limit is given by the value of capital depreciation.

7

Interaction of Multiplier and Accelerator

The definition of total expenditures in the economy is:

$$Y_t = C_t + I_t + G_t$$

where Y_t = sum of expenditures on new production during period t,
C_t = consumer expenditures during period t,
I_t = business expenditures during period t, and
G_t = government expenditures during period t.

It is assumed that:

$C_t = aY_{t-1}$ consumer spending is determined by income in the previous period and the value of the Propensity to Consume (a),

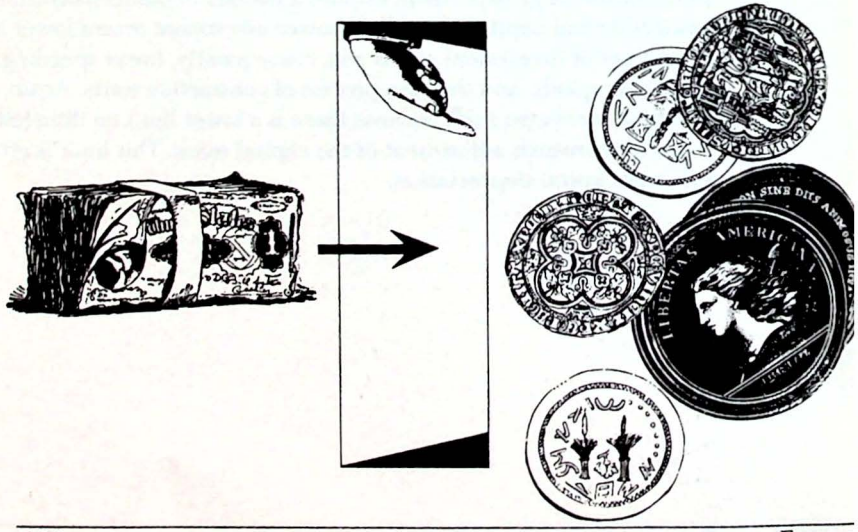
$I_t = b(C_t - C_{t-1})$ business investment depends on change in consumer spending from the previous period and the value of the accelerator (b), and

G_t is autonomous and for this model may include autonomous expenditures by the business sector.

Then, by substitution:

$$Y_t = a(1+b)Y_{t-1} - abY_{t-2} + G_t$$

The following program evaluates the model to find successive aggregate spending (outputs) for given values of the parameters of the equation. To keep the model within realistic levels, we can assume the maximum value of $Y = Y_m$ and the minimum value of $I = I_m$.



THE
PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A 1 − RCL 1 = × RCL 3 = STO 5 R/S	Calculates the value of G_t for stationary equilibrium.
014	2nd Lbl B RCL 1 × (1 + RCL 2) × RCL 3 = − RCL 1 × RCL 2 × RCL 4 + RCL 5 = 2nd Prt R/S 2nd Exc 3 2nd Exc 4 B	Calculates successive values of Y_t, Y_{t+1}, etc., with no limits on Y_m and I_m.
049	2nd Lbl C RCL 7 x⇄t RCL 3 − RCL 4 = × RCL 1 × RCL 2 = INV 2nd x≥t D	Calculates successive values of Y_t, Y_{t+1}, etc., with limits of Y_m and I_m. Begins by checking value of I_t.
070	2nd Lbl E + RCL 1 × RCL 3 + RCL 5 = STO 8 RCL 6 x⇄t RCL 8 2nd x≥t 2nd A'	If $I > I_m$, program generates Y_t and checks against Y_m.
091	2nd Lbl 2nd B' 2nd Prt R/S 2nd Exc 3 2nd Exc 4 C	If $Y_t < Y_m$, displays Y_t; then advances data in memories to calculate Y_{t+1}.
100	2nd Lbl D RCL 7 E	If $I_t < I_m$, program substitutes I_m and loops back to E.
105	2nd Lbl 2nd A' RCL 6 2nd Prt R/S 2nd Exc 3 2nd Exc 4 C	If $Y_t > Y_m$, program substitutes Y_m, advances data, and loops back to calculate Y_{t+1}.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER
CONTENTS

R_{00}		R_{05}	G_t
R_{01}	a	R_{06}	Y_m
R_{02}	b	R_{07}	I_m
R_{03}	Y_{t-1}	R_{08}	Used
R_{04}	Y_{t-2}	R_{09}	

7

Interaction of Multiplier and Accelerator

Using the Program

Procedure	Press	Display/Comments
Enter given data.	a STO 01	a
	b STO 02	b
	Y_{t-1} STO 03	Y_{t-1}
	Y_{t-2} STO 04	Y_{t-2}
	Y_m STO 06	Y_m
	I_m STO 07	I_m
Calculate G_t.	A	G_t
Calculate successive values of Y_t with no limits (continue to press R/S for successive values), or	B	Y_t° No limits.
	R/S	Y_{t+1}°
	R/S	Y_{t+2}°
Calculate successive values of Y_t with limits.	C	Y_t° With limits of
	R/S	Y_{t+1}° Y_m and I_m.
	R/S	Y_{t+2}°

°This value is printed if the PC-100A is used.

Sample Problem

Assume the following values:

$a = 0.9$	$Y_{t-1} = 1000$	$Y_m = 1500$
$b = 2$	$Y_{t+2} = 1000$	$I_m = -80$

First, solve for the equilibrium value of G_t. Next, assume a disturbance consisting of a permanent 20% increase in G_t, and find Y_t for six successive periods, first with no limits and then with limits given above.

Procedure	Press	Display/Comments
Enter program.		0
Enter data:		
a	.9 STO 01	0.9
b	2 STO 02	2.
Y_{t-1}	1000 STO 03	1000.
Y_{t-2}	1000 STO 04	1000.
Y_m	1500 STO 06	1500.
I_m	80 +/- STO 07	−80.

Procedure	Press	Display/Comments
Calculate equilibrium value of G_t.	A	100. Stored in memory 5.
Enter disturbance (20% increase).	X .2 =	20.
	SUM 05	20.
Calculate Y_t through Y_{t+5} with no limits.	B	1020.
	R/S	1074.
	R/S	1183.8
	R/S	1383.06
	R/S	1723.422
	R/S	2283.7314
Enter original Y_{t-1} and Y_{t-2} values and calculate Y_t through cycle *with* given limits. (Extend beyond 6 periods to see cyclical effect.)	1000 STO 03	1000.
	1000 STO 04	1000.
	C	1020.
	R/S	1074.
	R/S	1183.8
	R/S	1383.06
	R/S	1500.
	R/S	1500.
	R/S	1470.
	R/S	1389.
	R/S	1290.1
	R/S	1201.09

FURTHER EXPLORATIONS

The program can be used to show how different the cyclical development would be if the values of the parameters changed. Also, minor adjustments in the program can introduce a constant long-term growth rate. Finally, the model may be made more realistic by introducing a multiperiod adjustment by business to increased demand. This means that the induced investment will be spread over two or more periods in the future.

REFERENCES/ INFORMATION

As mentioned earlier, the first mathematical presentation of the combined effect of multiplier and accelerator was published in 1939. Since that time most textbooks on macroeconomics have included the model in various forms when discussing cyclical growth. See, for example, G. Ackley, *Macroeconomic Theory* (New York: The Macmillan Co., 1967), pp. 485-493.

7 Short-Run Production Function and Cost Curves

OBJECTIVE

In many courses covering microeconomic theory, the short-run cost of production is presented in the form of cost curves developed from a specific production function. Algebraic presentation requires not only some knowledge of calculus, but also a good understanding and ability to visualize the "shape" of a functional relationship. Very often much better understanding is obtained when a numerical example is shown. Of course, to present a numerical example – the derivation of cost curves from a production function, as well as the calculation of various measures of productivity – requires quite a bit of time.

A programmable calculator allows you to generate the successive values rapidly and thus grasp better how, for example, diminishing marginal product results in increasing marginal and average costs, or why average total cost may be U-shaped even when average variable cost never declines.

FACTS/ THEORY

In most textbooks, as well as in empirical research, an exponential function known as the Cobb-Douglas production function is used as an example of the input-output relationship that is characterized by the law of diminishing returns.* A standard form of the function may be as follows:

$$X = cL^aK^b$$

where X is the level of output, L and K are labor and capital inputs respectively, and a, b, and c are constants.

In our example we assume that L is the variable input, while K is the fixed input. Thus output can change only by adjusting employment of labor. This functional relationship between output and input is usually presented in the form of average and marginal product of labor, as well as average and marginal product of capital.

*See, for example, Edwin Mansfield, *Microeconomics: Theory and Applications* (New York: W. W. Norton and Co., 1970).

The respective concepts are:

Average product of Labor: $\frac{X}{L} = cL^{a-1}K^{b}$

Marginal product of Labor: $\frac{dX}{dL} = acL^{a-1}K^{b}$

Average product of Capital: $\frac{X}{K} = cL^{a}K^{b-1}$

Marginal product of Capital: $\frac{dX}{dK} = bcL^{a}K^{b-1}$

where
- a = output elasticity with respect to L,
- b = output elasticity with respect to K,
- c = technological constant,
- w = wage rate (price of L), and
- r = rate of return on capital (price of K),

while all are expressed as a function of L. It is clear that with a and b less than one (the expected case), the average and marginal products of labor will be declining as employment grows, while the average and marginal product of capital will grow for additional employment of labor.

When deriving cost curves, we assume for the purpose of exposition that input prices remain constant during the period in question. The total cost of production is defined as:

$$TC = wL + rK$$

Total cost is also thought of as consisting of variable cost (wL) and fixed cost (rK).

This simple definition of cost becomes slightly complicated when cost is to be described in relation to output rather than to inputs. The transformation of the function then requires inclusion of the production function, and variable cost becomes

$$VC = w\left(\frac{X}{cK^{b}}\right)^{\frac{1}{a}}$$

Additional information — usually presented in graphical form — that can also be derived from the production function are these cost curves:

Marginal cost: $\frac{dTC}{dX} = \frac{VC}{aX}$

Average Variable cost: $AVC = \frac{VC}{X}$

Average Fixed cost: $AFC = \frac{rK}{X}$

Average Total cost: $ATC = \frac{VC}{X} + \frac{rK}{X}$

7

Short-Run Production Function and Cost Curves

We have then ten analytical concepts — five production measures and five cost measures — with each of them varying with the level of employment or the level of output. The successive values of each of them can be generated quickly using the program described below, for any assumed values of the functional parameters.

The concepts:

Production: Total Output
Marginal Product of Labor
Average Product of Labor
Marginal Product of Capital
Average Product of Capital

Cost: Total Cost
Marginal Cost
Average Variable Cost
Average Fixed Cost
Average Total Cost

The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A RCL 5 × RCL 4 y^x RCL 1 × RCL 3 y^x RCL 2 = 2nd Op 24 2nd Prt R/S	Calculates total output (X). *Note:* 2nd Op 24 increments memory register 4 by 1.
021	2nd Lbl B RCL 1 ÷ RCL 4 × A	Calculates marginal product of labor (MPL).
030	2nd Lbl C RCL 4 $1/x$ × A	Calculates average product of labor (APL).
037	2nd Lbl D RCL 2 ÷ RCL 3 × A	Calculates marginal product of capital (MPK).
046	2nd Lbl E RCL 3 $1/x$ × A	Calculates average product of capital (APK).
053	2nd Lbl 2nd A' RCL 3 × RCL 7 + SBR CLR 2nd Prt R/S	Calculates total cost (TC).
065	2nd Lbl 2nd B' RCL 1 $1/x$ × 2nd C' R/S	Calculates marginal cost (MC).

Step #	Program Keystrokes	Purpose/Comments
073	2nd Lbl 2nd C' RCL 6 1/x × SBR CLR 2nd Prt R/S	Calculates average variable cost (AVC).
083	2nd Lbl 2nd D' RCL 3 × RCL 7 ÷ RCL 6 = 2nd Prt 2nd Op 26 R/S	Calculates average fixed cost (AFC). *Note:* 2nd Op 26 increments memory register 6 by 1.
098	2nd Lbl 2nd E' RCL 6 1/x × RCL 3 × RCL 7 + RCL 6 1/x × SBR CLR 2nd Prt R/S	Calculates average total cost (ATC).
118	2nd Lbl CLR ((RCL 5 × RCL 3 y^x RCL 2) 1/x × RCL 6) y^x RCL 1 1/x × RCL 8 = 2nd Op 26 INV SBR	Calculates variable cost (VC), which is included as a subroutine in calculations of TC, AVC, MC, and ATC.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}		R_{05}	c
R_{01}	a	R_{06}	X
R_{02}	b	R_{07}	r
R_{03}	K	R_{08}	w
R_{04}	L	R_{09}	

7

Short-Run Production Function and Cost Curves

Using the Program

Procedure	Press	Display/Comments
Set decimal for desired accuracy.	2nd fix n	(We'll use a 4-place decimal setting for convenience.)
Enter data in memory registers. *Note:* data *must* be entered in the designated memory registers.	a STO 01 b STO 02 K STO 03 L STO 04 c STO 05 X STO 06 r STO 07 w STO 08	a b K L c X r w
Production Analysis		
Calculate total output (X) based on original labor input (L).	A	X_1 ° Calculates first X and increments L by 1.
Calculate total output (X) based on incremented L.	A	X_2 °

Continue to press A to see effects on output caused by increments of 1 in labor input.

Enter original value of L and calculate marginal product of labor (MPL).	L STO 04 B	L MPL_1 ° Calculates first MPL and increments L by 1.
Calculate new MPL based on incremented L.	B	MPL_2 °

Continue to press B to see effects on MPL caused by increments of 1 in L.

Enter original value of L and calculate average product of labor (APL).	L STO 04 C	L APL_1 ° Calculates first APL and increments L by 1.
Calculate new APL.	C	APL_2 °

Continue to press C to see effects on APL caused by increments of 1 in L.

Enter original L and calculate marginal product of capital (MPK).	L STO 04 D	L MPK_1 °
	D	MPK_2 °

Continue to press D to see effects on MPK caused by increments of 1 in L.

Procedure	Press	Display/Comments
Enter original L and calculate average product of capital (APK).	L [STO] 04	L
	[E]	APK_1*
	[E]	APK_2*

Continue to press [E] to see effects on APK caused by increments of 1 in L.

Cost Analysis		
Calculate total cost.	[2nd] [A']	TC_1* Calculates total cost and increments X by 1.
Calculate TC based on incremented X (output value).	[2nd] [A']	TC_2*

Continue to press [2nd] [A'] to see effects on TC caused by increments of 1 in X.

Enter original value of X and calculate marginal cost (MC).	X [STO] 06	X
	[2nd] [B']	MC_1*
	[2nd] [B']	MC_2*

Continue to press [2nd] [B'] to see effects on MC caused by increments of 1 in X.

Enter original value of X and calculate average variable cost (AVC).	X [STO] 06	X
	[2nd] [C']	AVC_1*
	[2nd] [C']	AVC_2*

Continue to press [2nd] [C'] to see effects on AVC caused by increments of 1 in X.

Enter original X and calculate average fixed cost (AFC).	X [STO] 06	X
	[2nd] [D']	AFC_1*
	[2nd] [D']	AFC_2*

Continue to press [2nd] [D'] to see effects on AFC caused by increments of 1 in X.

Enter original X and calculate average total cost (ATC).	X [STO] 06	X
	[2nd] [E']	ATC_1*
	[2nd] [E']	ATC_2*

Continue to press [2nd] [E'] to see effects on ATC caused by increments of 1 in X.

*This value is printed if the PC-100A is used.

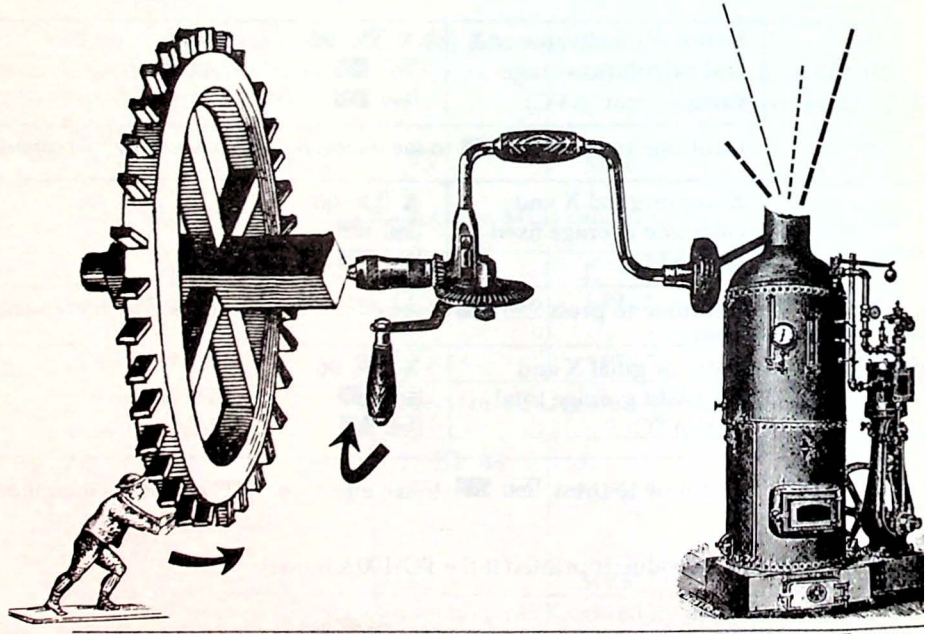
7

Short-Run Production Function and Cost Curves

Sample Problem

Assume a production function of $X = 1.2(L^{.7}K^{.3})$ with a wage rate (w) = 3, the price of capital (r) = 2, and a capital input (K) of 200. For production analysis, find total output (X), MPL, APL, MPK, and APK for labor inputs (L) of 300 through 304. For cost analysis, find TC, MC, AVC, AFC, and ATC for total outputs (X) of 320 through 324.

Procedure	Press	Display/Comments
Enter program.		0
Fix decimal.	2nd Fix 4	0.0000
Enter data in memories.		
(a)	.7 STO 01	0.7000
(b)	.3 STO 02	0.3000
(K)	200 STO 03	200.0000
(L)	300 STO 04	300.0000
(c)	1.2 STO 05	1.2000
(X)	320 STO 06	320.0000
(r)	2 STO 07	2.0000
(w)	3 STO 08	3.0000



Procedure	Press	Display/Comments
Production analysis:		
Calculate total outputs (X).		
L = 300	A	318.7683
L = 301	A	319.5117
L = 302	A	320.2544
L = 303	A	320.9963
L = 304	A	321.7376
Enter original L and calculate MPL for each value of L.	300 STO 04	300.0000
	B	0.7438
	B	0.7431
	B	0.7423
	B	0.7416
	B	0.7408
Enter original L and calculate APL.	300 STO 04	300.0000
	C	1.0626
	C	1.0615
	C	1.0604
	C	1.0594
	C	1.0583
Enter original L and calculate MPK.	300 STO 04	300.0000
	D	0.4782
	D	0.4793
	D	0.4804
	D	0.4815
	D	0.4826
Enter original L and calculate APK.	300 STO 04	300.0000
	E	1.5938
	E	1.5976
	E	1.6013
	E	1.6050
	E	1.6087

Procedure	Press	Display/Comments
Cost Analysis		
Calculate TC for X = 320 through 324.	2nd A'	1304.9720
	2nd A'	1309.0148
	2nd A'	1313.0630
	2nd A'	1317.1165
	2nd A'	1321.1754
Enter original X and calculate MC.	320 STO 06	320.0000
	2nd B'	4.0401
	2nd B'	4.0455
	2nd B'	4.0509
	2nd B'	4.0562
	2nd B'	4.0616
Enter original X and calculate AVC.	320 STO 06	320.0000
	2nd C'	2.8280
	2nd C'	2.8318
	2nd C'	2.8356
	2nd C'	2.8394
	2nd C'	2.8431
Enter original X and calculate AFC.	320 STO 06	320.0000
	2nd D'	1.2500
	2nd D'	1.2461
	2nd D'	1.2422
	2nd D'	1.2384
	2nd D'	1.2346
Enter original X and calculate ATC.	320 STO 06	320.0000
	2nd E'	4.0780
	2nd E'	4.0779
	2nd E'	4.0778
	2nd E'	4.0778
	2nd E'	4.0777

FURTHER EXPLORATIONS

The example presented here may also provide better understanding of the effects of change in the parameters of the production and cost functions. For example, you can study the effect of technological change simply by increasing the value of the constant (c) in the production function. Also, you may change the wage rate (w) to see the effects of wage increases on the various cost functions.

A production characterized by increasing returns to scale can easily be introduced by changing the values of a and b, the parameters of the production function, so that their sum is larger than 1. Also, the program can be redesigned for considerations of long-run production function and costs, as well as for different forms of production functions.

REFERENCES/ INFORMATION

A discussion of production functions and the derivation of cost curves is included in almost every text dealing with microeconomic theory.

8 Biology

Introduction

The historical approach to biology has been largely qualitative. Today's trend, however, is moving toward more and more quantitative measurement in various areas of biology. In these areas, mathematical models describing biological behavior are being set up and explored in ways that would have been virtually impossible before the development of our modern computational devices.

Your programmable calculator can be a powerful tool as you explore and discover for yourself the behavior of these mathematical descriptions and models. The programmable calculator can not only greatly reduce calculation time, but also allows you to search for reasonable alternative solutions by trial and error. Thus, you can examine the behavior of a model as its input parameters are changed. This type of analysis can give you a much more complete picture of all the possibilities of a given expression or set of equations describing an aspect of biological behavior.

The topics covered in this chapter are all areas which have seen increasing mathematical emphasis in recent years. Genetics was one of the first areas of biology to adopt a quantitative approach, and two examples from this field are included here. One is taken from the classical approach of Gregor Mendel. The other is a more advanced example involving blood-type matching. The study of population behavior is an area which is as quantitative a subject as any in biology today. Our example here is a simple one which only hints at the mathematical sophistication of models used in this field. Finally, since many of the problems in modern biology arise from the growing use of physical methods in research, the sedimentation of biological molecules has been chosen as a representative application for your programmable calculator.

In a chapter of this size, of course, we can only suggest the many possibilities and applications of your calculator. We hope you will explore other problems of your own choosing with the help of current literature on the subject of mathematics in biology. Two books of interest are listed below.

The Mathematical Approach to Physiological Problems, by Douglas Shepard Riggs. Cambridge, Mass.: M.I.T. Press, 1970. Chapters 5 through 11 provide many interesting problems in the advanced area of physiology.

Mathematical Ideas in Biology, by J. Maynard Smith. New York: Cambridge University Press, 1968. This book contains a great deal of material that can be handled by the programmable calculator.

8 Genetics

OBJECTIVE

In Mendelian genetics, dominance determines the character of the second generation zygotes. The program included here calculates the probabilities of the occurrence of combinations of traits in the second generation of a hybridized organism.

FACTS/ THEORY

Dealing with one dominant trait S, whose corresponding recessive trait is s, and assuming a genetically pure parent generation, the parent zygotes will be

SS and ss.

The first generation of hybrids will each have the zygotes Ss and will show the trait S. To examine the second generation, all possible genetic combinations must be considered. This can be done using a diagram called a Punnett Square, as shown here.

♂ Gametes \ ♀ Gametes	S	s
S	SS	Ss
s	Ss	ss

By the principle of dominance, all but the ss zygote will exhibit the dominant trait S. Therefore, for the second generation of mono-hybridized organisms, ¾ of the organisms display the dominant trait, and ¼ display the recessive trait.

Now consider two dominant traits, S and W, with their corresponding recessives s and w. If you wish to calculate the probability of finding the combination Sw in the second generation organisms, the formula is

$$P_{Sw} = P_S P_w$$

where P_S = the probability of finding S, and
P_w = the probability of finding w.

We have already established that the probability of finding a dominant trait is ¾ and the probability of finding a recessive trait is ¼ in the second generation of a mono-hybridized organism. Therefore, the probability of Sw is $¾ \times ¼ = {}^3/_{16}$, or, in general terms:

$$P_{ab\ldots n} = P_a P_{b\ldots} P_n$$

If we want the probability of a particular combination of m dominant traits for an n-hybrid, we use the formula

$$P_1^m = (3/4)^m(1/4)^{n-m} \quad \textit{(equation 1)}$$

The probability of an n-hybrid having m dominant traits and n − m recessive traits is given by the formula

$$P_2^m = (3/4)^m(1/4)^{n-m}\left[\frac{n!}{m!(n-m)!}\right] \quad \textit{(equation 2)}$$

The difference in the probabilities can be distinguished by a short example. Suppose we have a gamete with two dominant traits A and B and their corresponding recessive traits a and b. If we want the probability of a second generation zygote having the dominant trait A, we would use the first probability *(equation 1)*. Similarly, if we wanted the probability of the zygote having the dominant trait B, we would use *equation 1*. However, if we merely want to know the probability of the zygote having one dominant trait, either A or B but not both, we would use the second probability equation.

The following program uses both these expressions to explore the probabilities of dominant traits appearing in a second generation organism. Note that m = n at the beginning of the program, but m is decreased by one each time the program loops back to calculate a new probability.

THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 0 STO 1 STO 5 STO 6 2nd Op 25 4 1/x STO 3 × 3 = STO 02	Stores n, m, n + 1, probability of a recessive, and probability of a dominant.
021	2nd Lbl 2nd A' RCL 6 × 2nd Dsz 6 2nd A' 1 = STO 11 STO 12	Calculates and stores n! (and first m!)
035	2nd Lbl 2nd B' 0 x:t RCL 0 − RCL 1 = STO 8 2nd x=t 2nd D'	Calculates n − m and transfers to D', if result = 0.
049	2nd Lbl 2nd C' RCL 8 × 2nd Dsz 8 2nd C' 1 = STO 13 B	Calculates and stores (n − m)! if n − m ≠ 0.

Step #	Program Keystrokes	Purpose/Comments
062	2nd Lbl 2nd D° 1 STO 13	Stores 1 for (n − m)!, if n − m = 0.
067	2nd Lbl B RCL 2 y^x RCL 1 × RCL 3 y^x (RCL 0 − RCL 1) = STO 10	Calculates and stores $(\frac{3}{4})^m \times (\frac{1}{4})^{n-m}$ (probability of a particular combination of m dominant traits).
088	× RCL 11 ÷ RCL 12 ÷ RCL 13 = STO 04	Completes formula to find probability of all possible combinations of m dominant traits: $(\frac{3}{4})^m(\frac{1}{4})^{n-m}\left[\frac{n!}{m!(n-m)!}\right]$
100	RCL 1 2nd Prt R/S	Recalls m and stops.
104	RCL 10 2nd Prt R/S	Recalls probability of specific combination of m dominant traits, and stops.
108	RCL 4 2nd Prt 2nd Adv R/S	Recalls probability of all possible combinations of m dominant traits, and stops.
113	1 x:t RCL 1 INV 2nd x≥t E RCL 12 ÷ RCL 1 = STO 12	Checks to see if m < 1. If so, transfers to E. If m ≥ 1, calculates and stores new m!
128	2nd Lbl D 2nd Op 31 2nd Dsz 5 2nd Bst 1 +/− R/S	Decreases m by 1; checks register 5 to see if another loop is required (program loops n + 1 times). If not, displays −1 and stops.
138	2nd Lbl E 1 STO 12 D	Enters 1 for m! if m < 1; then transfers back to D.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}	n	R_{05}	Used	R_{10}	P_1
R_{01}	m	R_{06}	Used	R_{11}	n!
R_{02}	¾	R_{07}		R_{12}	m!
R_{03}	¼	R_{08}	Used	R_{13}	(n − m)!
R_{04}	P_2	R_{09}		R_{14}	

USING THE PROGRAM

Procedure	Press	Display/Comments
Enter number of traits (n).	n [A]	m_1° First number of dominant traits to be considered.
	[R/S]	P_1^m° Probability of finding all m dominant traits in second generation of n-hybridized organism.
	[R/S]	P_2^m° Probability of finding all possible combinations of m dominant traits.
	[R/S]	(m − 1)°
	[R/S]	P_1^{m-1}° Probability of finding specific combination of m − 1 dominant traits.
	[R/S]	P_2^{m-1}° Probability of finding any possible combination of m − 1 dominant traits.
Continue to press [R/S] until all probabilities have been calculated.	...	...
	[R/S]	0.° No dominant traits present.
	[R/S]	P_1^0° Probability of 0 dominant traits.
	[R/S]	P_2^0° Probability of 0 dominant traits.
	[R/S]	− 1. Signifies end of run.

°This value is printed if the PC-100A is used.

8

GENETICS

SAMPLE PROBLEM

Check the results for the di-hybridized system of sweet peas, considering the traits of stem length and seed shape. The traits are

S = long stems,
s = short stems,
W = round seeds, and
w = wrinkled seeds.

(The capital letters indicate dominance.)

Procedure	Press	Display/Comments
Enter program.		**0**
Enter number of traits being considered.	2 [A]	**2.** The number of dominant traits displayed (m).
	[R/S]	**0.5625** Probability of both dominant traits present.
	[R/S]	**0.5625** Probability of any possible combination of the two dominant traits.
	[R/S]	**1.** The number of dominant traits present (m − 1).
	[R/S]	**0.1875** The probability of finding S only and the probability of finding W only (P_1).
	[R/S]	**0.375** The probability of finding either S or W (P_2).
	[R/S]	**0.** No dominant traits displayed (m − 2 = 0).
	[R/S]	**0.0625** The probability of finding 0 dominant traits.
	[R/S]	**0.0625** The probability of finding neither S nor W.

Note that P_1 and P_2 are always equal where m = n and where m = 0. The information generated by the program may be charted as follows:

Symbol	*Traits*	*Probability 1*	*Probability 2*
SW	Long stem/round seeds	.5625	.5625
Sw	Long stem/wrinkled seeds	.1875	.375
sW	Short stem/round seeds	.1875	
sw	Short stem/wrinkled seeds	.0625	.0625

Check the results by making a Punnett Square for this system.

	SW	Sw	sW	sw
SW	SSWW	SSWw	SsWW	SsWw
Sw	SSWw	SSww	SsWw	Ssww
sW	SsWW	SsWw	ssWW	ssWw
sw	SsWw	Ssww	ssWw	ssww

Probability of SW = $^9/_{16}$ = .5625
Probability of Sw = $^3/_{16}$ = .1875
Probability of sW = $^3/_{16}$ = .1875
Probability of either Sw or sW = $^6/_{16}$ = .375
Probability of sw = $^1/_{16}$ = .0625

FURTHER EXPLORATIONS

Add a trait to the sweet pea problem; for example, seed color, where yellow seeds are dominant and green seeds are recessive. Determine the probabilities for all combinations of traits and confirm these results by making a Punnett Square.

FERENCES/ ORMATION

W. H. Johnson *et al, Biology,* 3rd ed. (New York: Holt, Rinehart and Winston, 1966), pp. 598-607.

Most introductory biology texts include a discussion of Mendelian and non-Mendelian genetics.

8 Blood-Type Matching

OBJECTIVE

The blood of an individual is typed by the presence or absence of the three antigens A, B, and Rh. The donation of blood does not require an exact match of blood types between donor and recipient, but the blood of the donor cannot have any antigens that are not present in the blood of the recipient. The following program examines a simple model of blood-type matching and generates the percentages of blood-type matches found in a given population.

FACTS/ THEORY

The presence of the A and/or B antigens is indicated by the letters A, B, or AB in the blood type, while the absence of both is indicated by the letter O. The + sign (positive) shows the presence of the Rh antigen, and the − sign (negative) its absence. Thus, type AB + means that an individual's blood has all three antigens; type O − indicates that none of the antigens is present. The following chart illustrates blood-type matching for donating or receiving blood.

Blood Type	*Recipients (Those who can receive blood type)*	*Donors (Those who can donate to blood type)*
A+	A+, AB+	A+, O+, A−, O−
B+	B+, AB+	B+, O+, B−, O−
AB+	AB+	A+, B+, AB+, O+, A−, B−, AB−, O−
O+	A+, B+, AB+, O+	O+, O−
A−	A+, AB+, A−, AB−	A−, O−
B−	B+, AB+, B−, AB−	B−, O−
AB−	AB+, AB−	A−, B−, AB−, O−
O−	A+, B+, AB+, O+, A−, B−, AB−, O−	O−

If you know the distribution of the eight blood types within a given populatio you can use the program included here to find the number and percentages of eligible donors and recipients for each blood type. For matching purposes in the program the blood types are given these numerical codes:

A+ = 1	AB+ = 3	A− = 5	AB− = 7
B+ = 2	O+ = 4	B− = 6	O− = 8

If you are using the TI Programmable 58, it will be necessary for you to repartition the data/program storage registers before entering the program, since it has 254 steps. Repartition by pressing **2** 2nd Op **17** before you enter the program. This allows you to enter up to 320 program steps, with 20 memories for data storage.

THE P'ROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 2nd Ind 0 SUM 10 2nd Op 20 R/S	Data-loading subroutine for the number of individuals having each blood type in the population.
009	2nd Lbl B STO 00 5 x:t RCL 0 2nd x=t x^2 RCL 0 x:t 1 2nd x=t D 2 2nd x=t E 3 2nd x=t 2nd E' RCL 4 + RCL 8 2nd D'	*Donor subroutines.* Tests the blood type being matched. If Rh factor is negative, program transfers to subroutine x^2. If positive, program tests for match. If blood type is 4 (0+), finds number and percentage of donors in the population.
037	2nd Lbl x^2 RCL 0 x:t 5 2nd x=t 2nd A' 6 2nd x=t 2nd B' 7 2nd x=t 2nd C' RCL 8 2nd D'	Tests Rh negative types and finds number and percentage of donors if blood type is 8 (O−).
054	2nd Lbl D RCL 1 + RCL 4 + RCL 5 + RCL 8 2nd D'	Donors for type 1 (A+).
068	2nd Lbl E RCL 2 + RCL 4 + RCL 6 + RCL 8 2nd D'	Donors for type 2 (B+).
082	2nd Lbl 2nd E' RCL 10 2nd D'	Donors for type 3 (AB+).
087	2nd Lbl 2nd A' RCL 5 + RCL 8 2nd D'	Donors for type 5 (A−).
095	2nd Lbl 2nd B' RCL 6 + RCL 8 2nd D'	Donors for type 6 (B−).
103	2nd Lbl 2nd C' RCL 5 + RCL 6 + RCL 7 + RCL 08	Donors for type 7 (AB−).

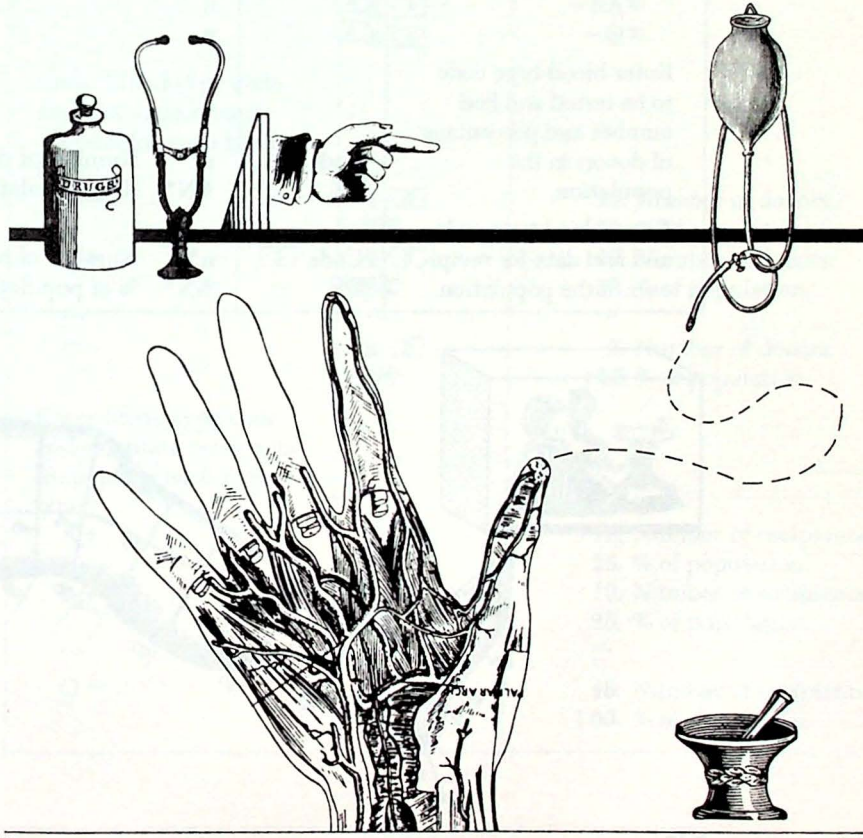
Step #	Program Keystrokes	Purpose/Comments
116	2nd Lbl 2nd D' = 2nd Prt R/S ÷ RCL 10 × 100 = STO 09	Finds total number and percentage of donors for blood type in question.
131	2nd Op 00 610000 2nd Op 04 RCL 9 2nd Op 06 R/S	Prints percentage and percent sign. If you are not using the PC-100A printer, this segment of the program may be omitted and replaced by a R/S .
146	2nd Lbl C STO 00 5 x:t RCL 0 2nd x=t y^x RCL 0 x:t 1 2nd x=t ln x 2 2nd x=t CE 3 2nd x=t CLR RCL 1 + RCL 2 + RCL 3 + RCL 4 2nd D'	*Recipient subroutines.* Tests the blood type being matched. If Rh factor is negative, transfers to subroutine y^x . If positive, program tests for match. If blood type is 4 (O+), finds number and percentage of recipients in the population.
180	2nd Lbl y^x RCL 0 x:t 5 2nd x=t $\sqrt{x}$ 6 2nd x=t 1/x 7 2nd x=t EE RCL 10 2nd D'	Tests Rh negative types and finds number and percentage of recipients if blood type is 8 (O−).
197	2nd Lbl ln x RCL 1 + RCL 3 2nd D'	Recipients for type 1 (A+).
205	2nd Lbl CE RCL 2 + RCL 3 2nd D'	Recipients for type 2 (B+).
213	2nd Lbl CLR RCL 3 2nd D'	Recipients for type 3 (AB+).
218	2nd Lbl $\sqrt{x}$ RCL 1 + RCL 3 + RCL 5 + RCL 7 2nd D'	Recipients for type 5 (A−).
232	2nd Lbl 1/x RCL 2 + RCL 4 + RCL 6 + RCL 8 2nd D'	Recipients for type 6 (B−).

Step #	Program Keystrokes	Purpose/Comments
246	2nd Lbl EE RCL 3 + RCL 7 2nd D'	Recipients for type 7 (AB−).
	LRN RST	Exits learn mode.

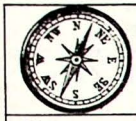
Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}	Used	R_{05}	#A−	R_{10}	total population (N)
R_{01}	#A+	R_{06}	#B−	R_{11}	
R_{02}	#B+	R_{07}	#AB−	B_{12}	
R_{03}	#AB+	R_{08}	#O−	R_{13}	
R_{04}	#O+	R_{09}	Used	R_{14}	



8 Blood-Type Matching



Using the Program

Procedure	Press	Display/Comments
Clear memories.	2nd CMs	
Enter 1 in memory 0.	1 STO 00	Directs program to begin data storage in memory 1.
Enter number of individuals with each blood type.		
#A+	A	n
#B+	A	n
#AB+	A	n
#O+	A	n
#A−	A	n
#B−	A	n
#AB−	A	n
#O−	A	n
Enter blood-type code to be tested and find number and percentage of donors in the population.	Code B R/S	n° %N°
Enter blood-type code and find data for recipients in the population.	Code C R/S	n° %N°

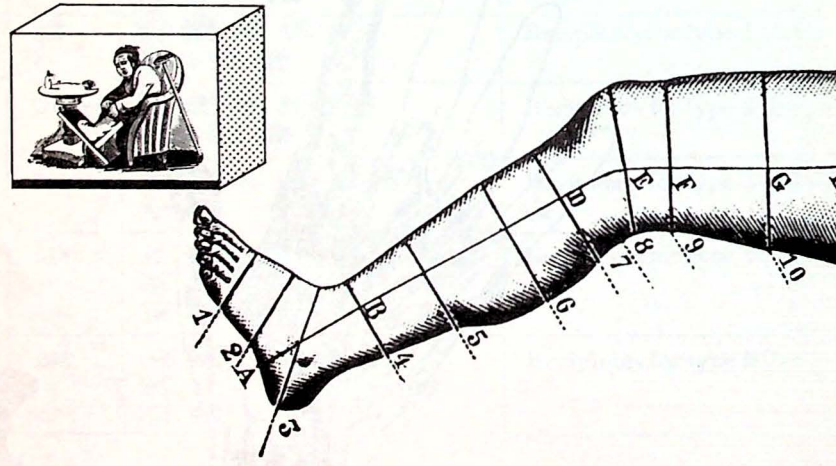
Note that data *must* be entered in this order.

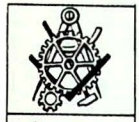
Number of donors.

% of population.

Number of recipients.

% of population.





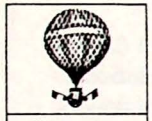
SAMPLE PROBLEM

Assume that all eight blood types are equally distributed within a population of forty individuals. Calculate the complete donor/recipient table for each blood type.

Procedure	Press	Display/Comments
Enter program.		0
Clear memories.	2nd CMs	0.
Enter 1 in memory 0.	1 STO 00	1.
Enter data.	5 A	5.
	5 A	5.
	5 A	5.
	5 A	5.
	5 A	5.
	5 A	5.
	5 A	5.
	5 A	5.
	5 A	5.
Enter blood-type code and calculate donors. (Repeat for each blood type.)		
A+	1 B	20. Number of donors.
	R/S	50. % of population.
B+	2 B	20. Number of donors.
	R/S	50. % of population.
...	...	...
O−	8 B	5. Number of donors.
	R/S	12.5 % of population.
Enter blood-type code and calculate recipients. (Repeat for each blood type.)		
A+	1 C	10. Number of recipients.
	R/S	25. % of population.
B+	2 C	10. Number of recipients.
	R/S	25. % of population.
...	...	...
O−	8 C	40. Number of recipients.
	R/S	100. % of population.

The complete donor/recipient table will be:

Blood Type	*Donors*	*Recipients*
A+	20	10
	50%	25%
B+	20	10
	50%	25%
AB+	40	5
	100%	12.5%
O+	10	20
	25%	50%
A−	10	20
	25%	50%
B−	10	20
	25%	50%
AB−	20	10
	50%	25%
O−	5	40
	12.5%	100%



FURTHER EXPLORATIONS

It is not realistic, of course, to assume that all blood types occur with equal probability throughout the general population. The actual distribution is:

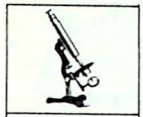
A+	37.7%	A−	6.3%
B+	8.5%	B−	1.5%
AB+	3.4%	AB−	0.6%
O+	37.4%	O−	6.6%

Using these figures, calculate a blood donor/recipient table for a population of 200.

REFERENCES/ INFORMATION

Dorn and McCracken, *Elements of Finite Mathematics with Computing* (New York: John Wiley and Sons, Inc., 1976), pp. 193-198.

Population Growth



OBJECTIVE

The growth of a population can be estimated by setting up a simple nonlinear model, with any parameter values which would be of interest in a specific problem. The following program allows you to study the behavior of such a model.

FACTS/ THEORY

Any model we use must make certain basic assumptions about the growth of a population. For example, here we will assume that some number (B) of individuals are born each breeding season and

$$B = aP_r$$

where B = the number of births each season,
a = a given positive constant, and
P_r = the population at the beginning of season r.

In other words, we assume that the number of births is just proportional to the size of the population at the beginning of the breeding season. On the other hand, the fraction of the population that dies during each season will not be constant, but increases as the population grows (due to scarcity of food, for instance). We will assume here that the fraction of the population that dies in a season is itself directly proportional to the population, or is equal to bP_r, and the number of deaths that occur during a season is

$$D = bP_r(P_r)$$

where b is a constant factor.

Thus the total population at the end of the season is expressed as

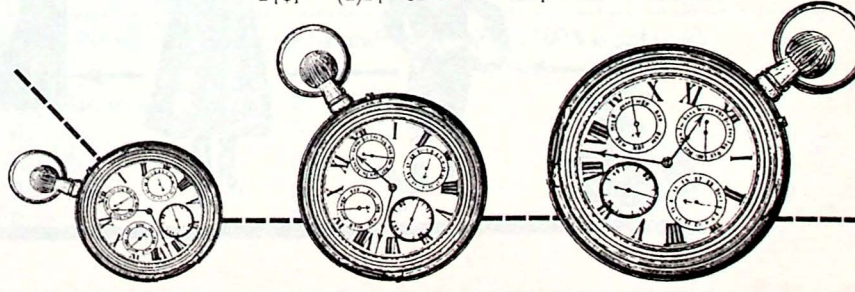
$$P_{r+1} = P_r + aP_r - bP_r(P_r)$$

or

$$P_{r+1} = (1 + a - bP_r)P_r$$

It also may happen that the population remains at a non-zero constant value after a number of seasons. That is, the population reaches some equilibrium number (P_{eq}) so that

$$P_{r+1} = (1)P_r \text{ or } 1 + a - bP_r = 1$$



Under these conditions, P_r may be called the equilibrium population P_{eq}, and

$$P_{eq} = \frac{a}{b}$$

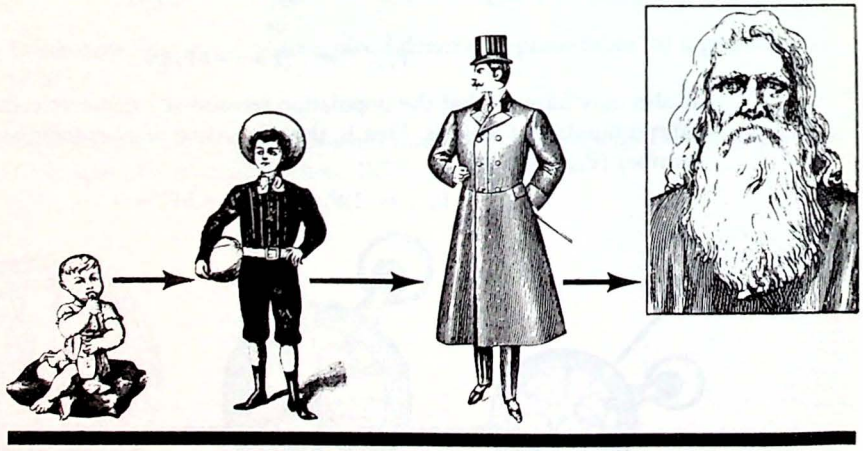
The population will eventually reach this value independent of the starting value of the population in some cases. Note that, if $P_r \geq \frac{a+1}{b}$, P_{r+1} is negative or zero, which must be interpreted to mean that the population is wiped out (it cannot, obviously, become negative). Therefore, an initial value of $P_0 \geq \frac{a+1}{b}$ is not a valid starting point for this model.

The program included here performs several functions. First, it solves the difference equation for the growth of a population and finds the equilibrium population, using inputs of a, b, and initial population. Keeping the above observations in mind, investigation of this equation for different values of a and b helps us to understand and follow the behavior of the model.

Also, if you know the values of P_0, P_1, and P_2, the program will calculate the constants a and b:

$$a = \frac{P_1^3 - P_0^2 P_2}{P_0 P_1 (P_1 - P_0)} - 1$$

$$b = \frac{P_1^2 - P_0 P_2}{P_0 P_1 (P_1 - P_0)}$$



THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 1 R/S STO 2 R/S STO 3 R/S	Enters a, b, and P_r (in that order).
011	2nd Lbl B 0 STO 0 RCL 1 ÷ RCL 2 = 2nd Prt 2nd Adv R/S	Calculates P_{eq}.
025	2nd Lbl C 1 + RCL 1 − RCL 2 × RCL 3 = 2nd Prd 3 2nd Op 20 RCL 0 2nd Prt R/S RCL 3 2nd Prt 2nd Adv R/S GTO C	Calculates r and P_{r+1}.
053	2nd Lbl 2nd A' STO 3 R/S STO 4 R/S STO 5 R/S	Enters P_r, P_{r+1}, and P_{r+2} (in that order).
064	2nd Lbl D INV 2nd Fix RCL 3 × RCL 4 × (RCL 4 − RCL 3) = STO 6 (RCL 4 y^x 3 − RCL 3 x^2 × RCL 5) ÷ RCL 6 − 1 = STO 1 2nd Prt R/S	Calculates a.
107	(RCL 4 x^2 − RCL 3 × RCL 5) ÷ RCL 6 = STO 2 2nd Prt R/S GTO B	Calculates b, then transfers to B to calculate P_{eq}.
	LRN RST	Exits learn mode.

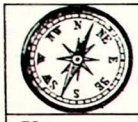
Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}	Counter	R_{05}	P_{r+2}
R_{01}	a	R_{06}	Used
R_{02}	b	R_{07}	
R_{03}	P_r	R_{08}	
R_{04}	P_{r+1}	R_{09}	

8

Population Growth



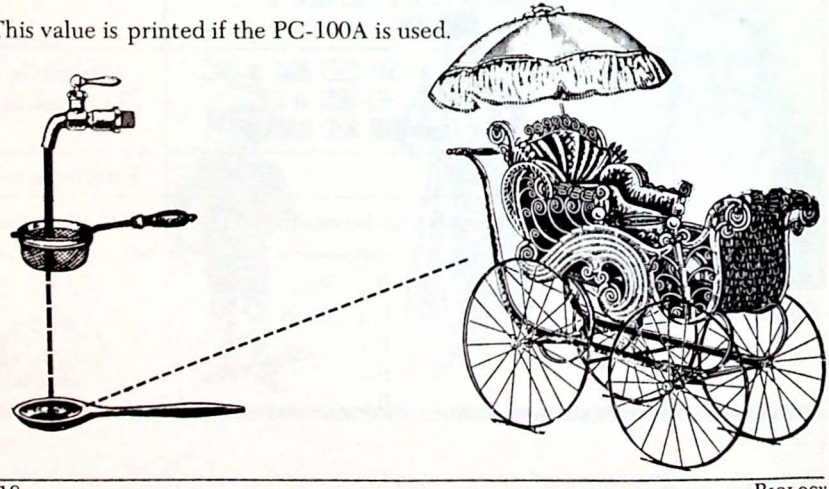
Using the Program

Procedure	Press	Display/Comments
To calculate P_{eq}, P_1, P_2, ... P_{eq}:		
Enter a.	A	a
Enter b.	R/S	b
Enter P_0.	R/S	P_0
Calculate P_{eq}.	B	P_{eq} °
Select decimal for desired accuracy.	2nd fix n	P_{eq} in set-decimal format.
Calculate P_1.	C	1.00° season 1 (r_1).
	R/S	P_1 °
Calculate P_2.	R/S	2.00°
	R/S	P_2 °

Continue to press R/S until population reaches equilibrium.

To calculate a, b, and P_{eq} (if P_0, P_1, and P_2 are known):		
Enter P_0, P_1, and P_2.	2nd A'	P_0
	R/S	P_1
	R/S	P_2
Compute a.	D	a°
Compute b.	R/S	b°
Calculate P_{eq}.	R/S	P_{eq} °

°This value is printed if the PC-100A is used.



SAMPLE PROBLEM

Calculate equilibrium population and population growth up to equilibrium, given the following information:

$a = 1.5$ $P_0 = 1000$
$b = .0001$

Procedure	Press	Display/Comments
Enter program.		0
Enter data.		
$a = 1.5$	1.5 [A]	1.5
$b = .0001$	.0001 [R/S]	0.0001
$P_0 = 1000$	1000 [R/S]	1000.
Calculate P_{eq}.	[B]	15000. P_{eq}
Set decimal for desired accuracy (we'll use 2-decimal format).	[2nd] [fix] 2	15000.00
Calculate population growth.	[C]	1.00 Season 1.
	[R/S]	2400.00 P_1.
	[R/S]	2.00 Season 2.
	[R/S]	5424.00 P_2.

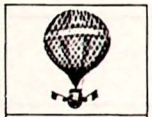
Continue to press [R/S] until population reaches equilibrium.

	...	...
	[R/S]	19.00 Season 19.
	[R/S]	14999.99 P_{19}.
	[R/S]	20.00 Season 20.
	[R/S]	15000.00 P_{20}.
	[R/S]	21.00 Season 21.
	[R/S]	15000.00 P_{21}.

The information generated will be as follows:

1.00	8.00	15.00
2400.00	15017.45	14999.86
2.00	9.00	16.00
5424.00	14991.24	15000.07
3.00	10.00	17.00
10618.02	15004.37	14999.97
4.00	11.00	18.00
15270.82	14997.81	15000.02
5.00	12.00	19.00
14857.26	15001.09	14999.99
6.00	13.00	20.00
15069.33	14999.45	15000.00
7.00	14.00	21.00
14964.85	15000.27	15000.00

Note that in this model the population oscillates about the equilibrium value until it stabilizes.



FURTHER EXPLORATIONS

1. Run the program with the same values for b and P_0, but increase a to 1.9. What is the effect of increasing a?

2. A simple model for finding the increase in the number of infected individuals in an epidemic is

$$I_{k+1} = (1 + CN - CI_k)I_k$$

where C = the constant rate,
N = the total population, and
I_k = the number of infectives at time $k\Delta t$.

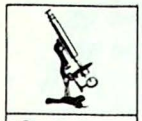
By letting $a = CN$ and $b = C$ and substituting I_k for P_0, the program in this section can be used to model the growth of the epidemic. Use the following values to run the program:

$$N = 1000 \qquad I_0 = 10$$
$$C = .0005$$

REFERENCES/ INFORMATION

Dorn and McCracken, *Elements of Finite Mathematics with Computing* (New York: John Wiley and Sons, Inc., 1976), chapters 3 and 5.

Sedimentation



OBJECTIVE

A centrifuge is used to separate elements of a solution by spinning a container of the solution at a rate on the order of 50,000 rpm at a distance of 5 to 10 cm from the axis of rotation. This rotation will subject the particles to a rather large centrifugal force and one might expect them to move very quickly. However, the solution also is subject to a velocity-dependent frictional force with the solvent, which causes the particle velocities to be very slow. The objective of the following exercise is to find a simple model of particle transport in a centrifuge, and examine the behavior of the sedimentation velocity up to equilibrium.

FACTS/ THEORY

The velocity of importance here is the sedimentation velocity. This is the rate at which the boundary between solute and solvent regions moves. In this simple model of particle transport, we will consider the motion of a single molecule and use its velocity to represent the sedimentation velocity. The forces acting upon a particle in a centrifuge are:

(1) the centrifugal force, $F_c = m\omega^2 r$,

where m = mass,
ω = the angular velocity, and
r = the distance from the center of rotation.

(2) the frictional force, $F_f = -fv$

where f = the coefficient of friction, and
v = velocity.

So the equation of motion is

$$m\ddot{r} = m\omega^2 r - fv,$$

or multiplying by Avogadro's number, N_o,

$$\ddot{r} = \omega^2 r - \frac{N_o f}{M} v$$

where M = molecular weight. Given the results of diffusion experiments, we can calculate the friction coefficient (f) from the measured diffusion coefficient as

$$N_o f = RT/D$$

where R = the universal gas constant,
T = the absolute temperature, and
D = the diffusion coefficient.

The diffusion coefficient is dependent on the temperature, but here we will always use diffusion coefficients given for room temperature, 20°C. So we may now write the equations of motion as

$$\ddot{r} = \omega^2 r - \frac{RT}{MD} v$$

Since r is nearly constant, the velocity will increase to some value where $\ddot{r} = 0$ and then remain constant. The constant value will be

$$v_{eq} = \frac{M\omega^2 r D}{RT}$$

In the following exercise, examine the behavior up to equilibrium and observe that the sedimentation velocity approaches this constant value. The program solves the equation of motion numerically, using the trapezoidal rule.

Consider the first order differential equation

$$y' = f(y,t)$$

with the boundary condition $y(t_o) = y_o$.

Given a uniform time mesh,

$$t_n = t_o + nh, \quad y_n = y(t_n), \quad f_n = f(y_n, t_n),$$

then the forward difference approximation to y'_n is

$$y'_n = (y_{n+1} - y_n)/h = f_n$$
$$y_{n+1} = y_n + hf_n$$

This rule may be used to predict a value of y_{n+1}. This predictor is used to calculate f_{n+1} so that the value of y_{n+1} may be corrected using the trapezoidal rule,

$$y_{n+1} = y_n + \frac{h}{2}(f_{n+1} + f_n)$$

This corrector may be applied repeatedly until y_{n+1} is constant to within some desired accuracy.

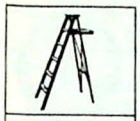
Since the sedimentation equation is second order, we must use this procedure on two coordinates, y_1 and y_2, where

$$y_1 = r$$
$$y_2 = v$$

so that

$$\dot{r} = f = y_2$$

$$\dot{v} = g = \omega^2 y_1 - \frac{RT}{MD} y_2$$



THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 4 STO 5 INV EE R/S STO 6 R/S STO 07 0 STO 8 R/S	Stores h, t_o, r, and $\dot{r}_o$ (= 0). *Note:* All data must be entered in the order shown here.
018	× 2 × 2nd π ÷ 60 = x^2 STO 14 R/S	Converts ω (rpm) to rad/sec and stores ω^2.
030	× R/S = 1/x STO 15 R/S	Computes MD and stores as 1/MD.
037	× 8.31 EE 7 = 2nd Prd 15 INV EE	Calculates RT ($R = 8.31 \times 10^7$); then calculates and stores RT/MD.
049	RCL 14 × RCL 7 ÷ RCL 15 = 2nd Prd R/S	Calculates equilibrium velocity.
060	STO 00 2 INV 2nd Prd 4 C RCL 4 2nd Prd 11 2nd Prd 12	Enters number of integrations you want to perform; then calculates h/2 and loops to C to calculate f and g. Then calculates $f_{n+1} = (h/2)f_n$ and $g_{n+1} = (h/2)g_n$.
073	2nd Lbl B 2nd Fix 5 RCL 5 SUM 6 RCL 7 STO 9 RCL 11 SUM 9 × 2 = SUM 7 RCL 8 STO 10 RCL 12 SUM 10 × 2 = SUM 08	*Integration Step.* Increments t by h and calculates predictor for $(y_1)_{n+1}$ and $(y_2)_{n+1}$.
107	2nd Lbl D C RCL 4 2nd Prd 11 2nd Prd 12 RCL 9 + RCL 11 = INV SUM 7 2nd Exc 7 2nd Pause STO 13	Transfers to C and returns. Multiplies f and g by h/2 to find and store corrected $(y_1)_{n+1}$ value.
130	RCL 10 + RCL 12 = INV SUM 8 2nd Exc 08	Calculates and stores corrected $(y_2)_{n+1}$ value.

8

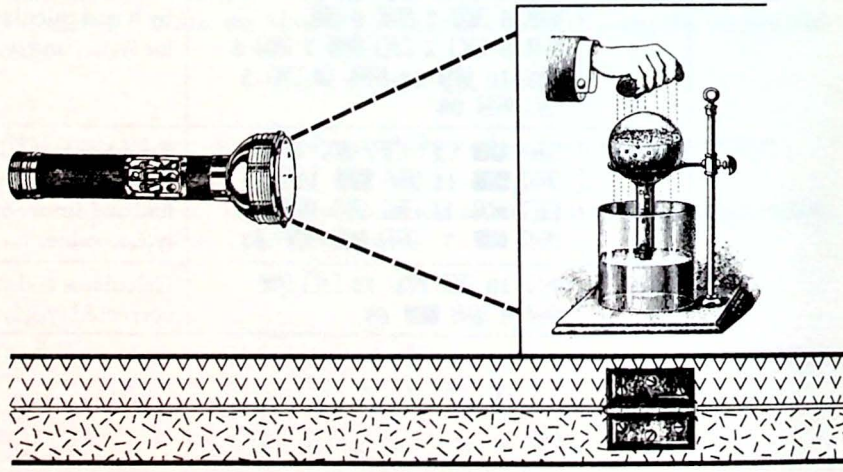
Sedimentation

Step #	Program Keystrokes	Purpose/Comments
141	2nd Ix1 + RCL 13 = EE INV EE INV 2nd x=t D INV 2nd Fix RCL 6 2nd Adv 2nd Prt R/S RCL 7 2nd Prt R/S RCL 8 2nd Prt R/S	Calculates total change in y_1 and y_2. If error $\neq 0$, loops back to D. If error $= 0$, recalls t, r, and v and stops.
167	2nd Dsz 0 B R/S	Checks memory 0 and repeats integration step, if required.
171	2nd Lbl C RCL 8 STO 11 RCL 14 X RCL 7 − RCL 15 X RCL 8 = STO 12 INV SBR	Calculates f(y,t) and g(y,t). $f = \dot{r}$ $g = \omega^2 r - (RT/MD)\dot{r}$
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

Register Contents

R_{00}	step counter	R_{05}	h	R_{10}	Used	R_{15}	RT/MD
R_{01}		R_{06}	t_{n+1}	R_{11}	Used	R_{16}	
R_{02}		R_{07}	r_{n+1}	R_{12}	Used	R_{17}	
R_{03}		R_{08}	$\dot{r}_{n+1}$	R_{13}	Used	R_{18}	
R_{04}	h/2	R_{09}	Used	R_{14}	ω^2	R_{19}	



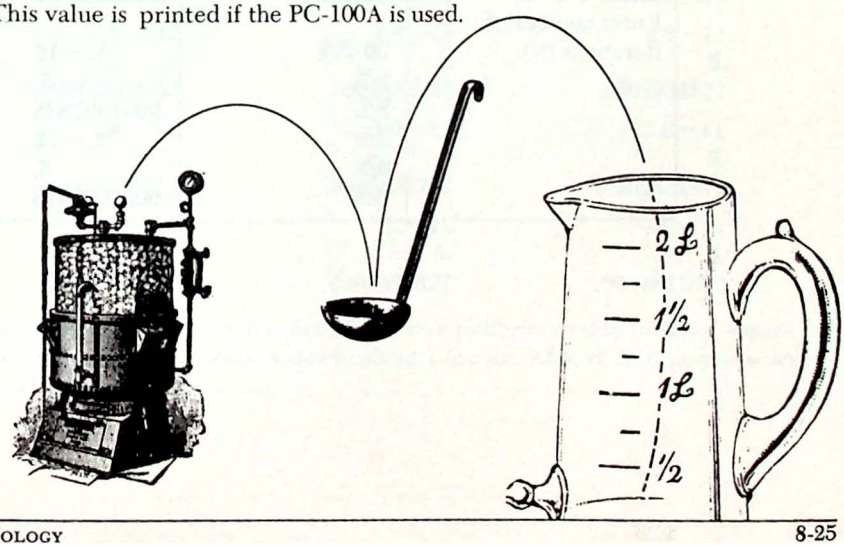
USING THE PROGRAM

Procedure	Press	Display/Comments
Enter step size (h).	h [A]	h Data-loading section.
Enter t_o.	t_o [R/S]	t_0
Enter r_o.	r_0 [R/S]	r_0
Enter speed of centrifuge in rpm.	ω [R/S]	ω^2 in rad/sec
Enter M.	M [R/S]	M
Enter D.	D [R/S]	1/MD
Enter T in °K.	T [R/S]	v_{eq}° Equilibrium velocity.

To observe the behavior up to equilibrium, enter the number of iterations you want to observe and press [R/S] to see the velocity at time increments of h.

	n [R/S]	t_1°
	[R/S]	r_1°
	[R/S]	v_1°
	[R/S]	t_2°
	[R/S]	r_2°
	[R/S]	v_2°
	[R/S]	...
	[R/S]	t_n°
	[R/S]	r_n°
	[R/S]	v_n°

°This value is printed if the PC-100A is used.



8

Sedimentation

Sample Problem

Given the following information, calculate the sedimentation velocity of bovine serum albumin in a centrifuge with a 5.0 cm radius spinning at 56,000 rpm at room temperature ($T = 20°C = 293°K$).

M = 66,500 g/mole
D = 0.603×10^{-6} cm²/sec
t_o = 0
h = 1×10^{-12} sec (1$\mu\mu$sec)
N = 30 steps

Procedure	Press	Display/Comments
Enter program.		0
Enter step size (h).	1 EE 12 +/− A	1. − 12
Enter starting time (t_o).	0 R/S	0.
Enter radius of centrifuge (r).	5 R/S	0.
Enter speed of centrifuge in rpm (ω).	56000 R/S	34390088.22 ω^2 in rad/sec.
Enter M.	66500 R/S	66500.
Enter D.	.603 EE 6 +/− R/S	2.4937967 01 1/MD
Enter T in °K.	293 R/S	.0002831872 v_{eq}
Enter number of iterations (N).	30 R/S	1. − 12 t_1
	R/S	5. r_1 (y_1)
	R/S	.0001307838 v_1 (y_2)
	R/S	2. − 12 t_2
	R/S	5. r_2
	R/S	.0002000978 v_2

Continue to press R/S and watch the changes as the velocity approaches equilibrium. The following information will be generated:

1. – 12	1.1 – 11	2.1 – 11
5.	5.	5.
.0001307838	.0002827201	.0002831845
2. – 12	1.2 – 11	2.2 – 11
5.	5.	5.
.0002000978	.0002829152	.0002831856
3. – 12	1.3 – 11	2.3 – 11
5.	5.	5.
.0002392481	.0002830211	.0002831862
4. – 12	1.4 – 11	2.4 – 11
5.	5.	5.
.0002614364	.0002830898	.0002831866
5. – 12	1.5 – 11	2.5 – 11
5.	5.	5.
.0002726789	.0002831281	.0002831869
6. – 12	1.6 – 11	2.6 – 11
5.	5.	5.
.0002773222	.0002831524	.000283187
7. – 12	1.7 – 11	2.7 – 11
5.	5.	5.
.0002794737	.0002831661	.0002831871
8. – 12	1.8 – 11	2.8 – 11
5.	5.	5.
.0002810753	.0002831747	.0002831871
9. – 12	1.9 – 11	2.9 – 11
5.	5.	5.
.0002818714	.0002831797	.0002831871
1. – 11	2. – 11	3. – 11
5.	5.	5.
.0002824287	.0002831827	.0002831872

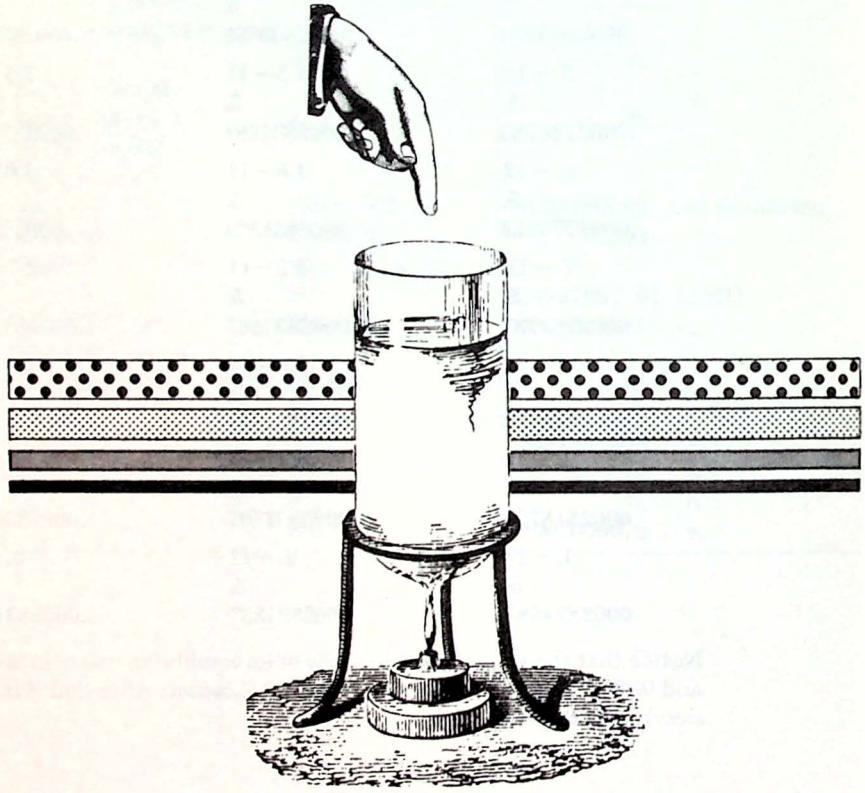
Notice that the velocity reaches 90% of its equilibrium value in about 4$\mu\mu$sec and 99% of its equilibrium value in about 7.5$\mu\mu$sec. After that it increases very slowly to its constant value.

FURTHER EXPLORATIONS

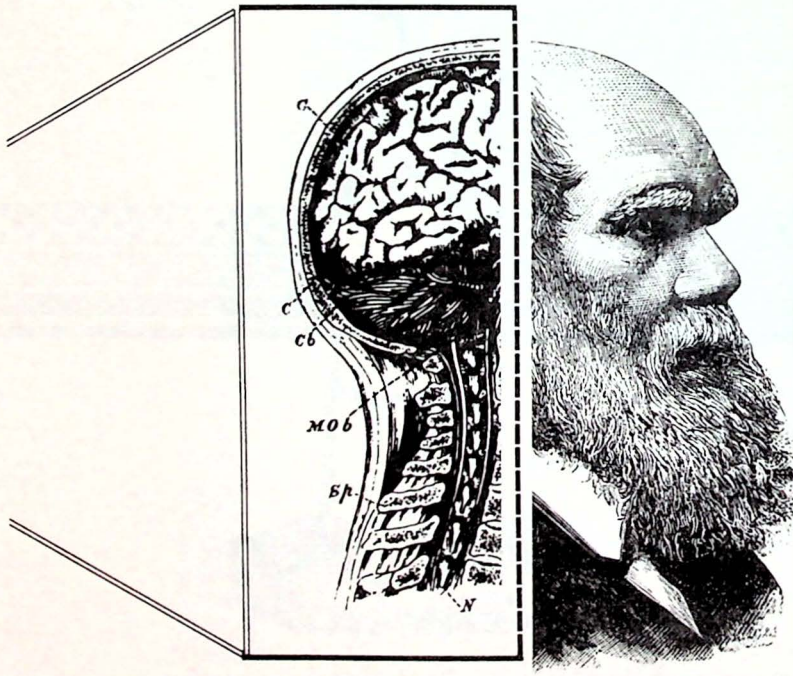
1. Find the sedimentation velocity for serum albumin if $r_o = 7$ cm and the centrifuge is set for 40,000 rpm.
2. For sucrose ($M = 342$, $D = 4.586 \times 10^{-6}$ cm^2/sec) at 20°C, find the value of ω in rpm that would give a sedimentation velocity of 1 cm/hr in a 7 cm centrifuge. (Note that the formulas used here will give ω in radians/sec. *Watch your units.*)

REFERENCES/ INFORMATION

K. Van Holde, *Physical Biochemistry* (Englewood Cliffs, N.J.: Prentice-Hall, 1971), pp. 79-97.



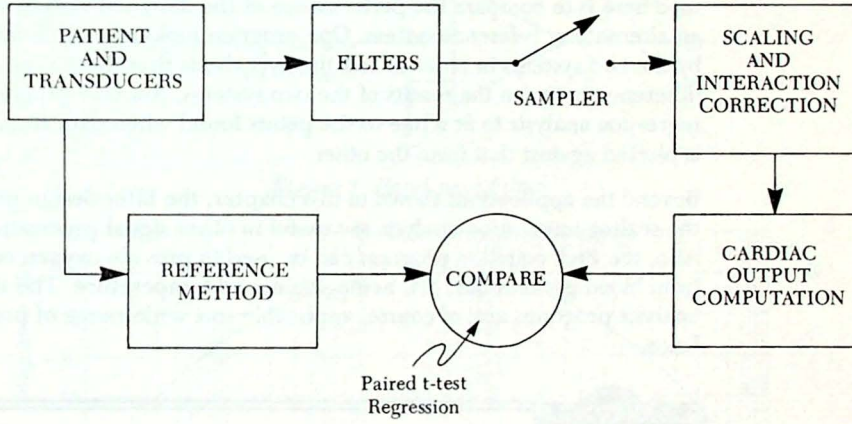
9 Biomedical Engineering



Introduction

Along with showing you several applications for your programmable calculator in biomedical engineering, this chapter also has another special purpose. It demonstrates the way these applications fit together into a complete project. Although each of the programs is useful in and by itself — and by no means limited to the specific usage shown here — their power is fully utilized when they operate in combination.

The ultimate objective in this chapter is to determine cardiac output by analyzing samples of arterial and venous blood and measuring body oxygen consumption. First, we need to design and optimize a means of collecting the necessary data. Next, the actual calculations for cardiac output must be made. Then the results need to be verified, and the system design evaluated. The elements of the system we use here are shown below.

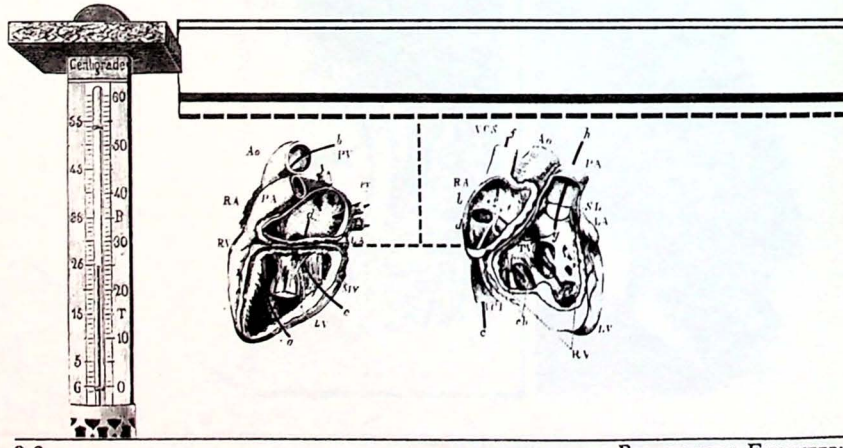


9 Introduction

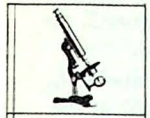
The patient and the transducers are one block of the system, from which electrical signals are obtained. The signals are filtered to improve the signal-to-noise ratio, and the calculator itself helps in the filter design. Next, the possibility of automated multiple measurements (to reduce variability by averaging) is considered. The calculator is used to relate accuracy after averaging to the number of samples averaged. Also, as part of the design consideration, we examine the possibility of using one transducer for all required measurements of oxygen gas tension. (Readings are taken sequentially.) In doing so, certain procedures are used to offset undesired interactions among sequential measurements caused by transient response characteristics of the measuring electrode. After measurements are taken, the cardiac output is calculated by the Fick equation. As part of the calculation, the oxygen content in the blood samples is determined.

Finally, we analyze the performance of the overall system. The approach used here is to compare the performance of the designed system with that of an alternate or reference system. One program looks at paired data obtained by the two systems in order to test the hypothesis that there is no significant difference between the results of the two systems. Another program uses regression analysis to fit a line to the points found when data from one system is plotted against that from the other.

Beyond the applications shown in this chapter, the filter design program and the scaling interaction analysis are useful in other signal processing problems. Also, the Fick equation program can be used to provide oxygen content data from blood gas tensions, pH, hemoglobin, and temperature. The statistical analysis programs are, of course, applicable to a wide range of problems and fields.



Active Filter to Improve Signal-to-Noise Ratio



OBJECTIVE

Transducer outputs are often observed to be noisy, which can make it difficult to accurately measure the signal output of a transducer. An effective means for improving signal-to-noise ratio is the use of a filter. The following program allows you to calculate the parameters of a simple operational amplifier active filter given initial design specifications.

FACTS/ THEORY

A circuit diagram for a simple band-pass filter using one operational amplifier is shown in *Figure 1*, and a representative frequency diagram for such a filter is shown in *Figure 2*.

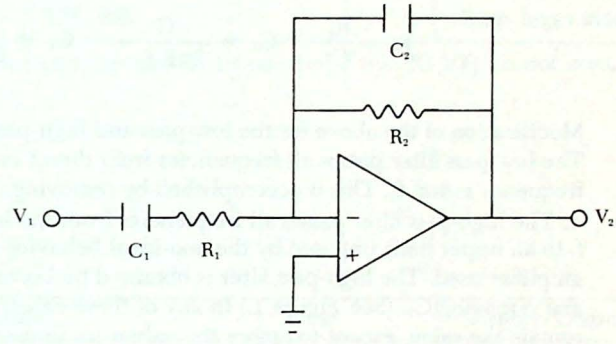


Figure 1. Band-pass Filter.

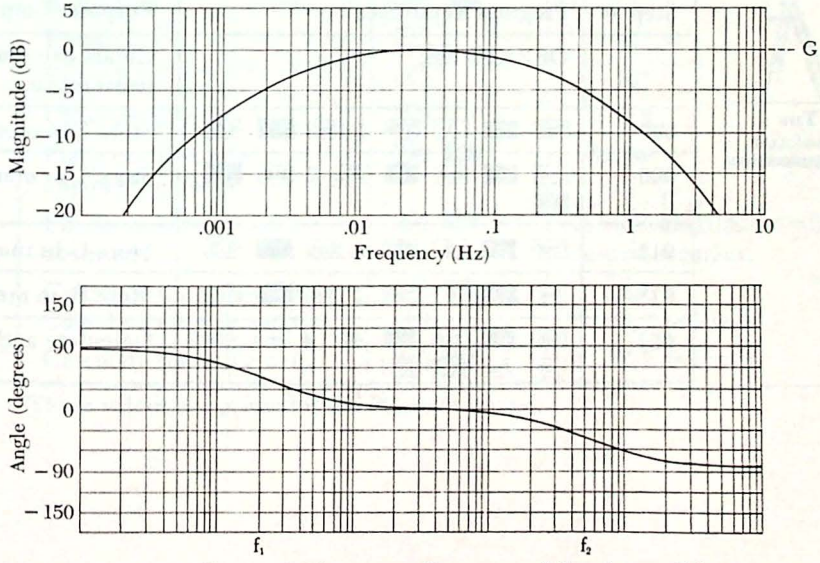


***Figure 2.** Frequency Response of Band-pass Filter.*

9

Active Filter to Improve Signal-to-Noise Ratio

The filter attenuates signal frequencies outside the passband. In *Figure 2*, the range of frequencies from f_1 to f_2 is the filter passband. The 3dB low-frequency cutoff is labeled f_1 and the 3dB high-frequency cutoff is labeled f_2. The phase shift in the passband is close to zero but approaches $\pm 45°$ at the cutoff frequencies.

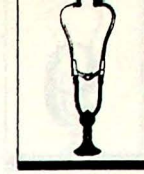
Inputs to the calculator are the gain (G) of the filter in the passband, the feedback resistance (R_2), and the 3dB cutoff frequencies (f_1 and f_2). The remaining parameter values are calculated as follows:

$$R_1 = \frac{R_2}{G} \qquad C_1 = \frac{G}{2\pi R_2 f_1} \qquad C_2 = \frac{1}{2\pi R_2 f_2}$$

Modification of the above for the low-pass and high-pass filter is very easy. The low-pass filter passes all frequencies from direct current to the high-frequency cutoff f_2. This is accomplished by removing capacitor C_1 in *Figure 1*. The high-pass filter passes all frequencies from the low-frequency cutoff f_1 to an upper limit imposed by the non-ideal behavior of the operational amplifier used. The high-pass filter is obtained by leaving C_1 in the circuit and removing C_2. (See *Figure 1.*) In any of these cases, the calculations remain the same, except to ignore the values for unused capacitors.

The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [A] [STO] 1 [2nd] [Prt] [R/S]	Store f_1 in memory 01.
006	[2nd] [Lbl] [2nd] [A'] [STO] 2 [2nd] [Prt] [R/S]	Store f_2 in memory 02.
012	[2nd] [Lbl] [B] [STO] 3 [2nd] [Prt] [R/S]	Store G in memory 03.
018	[2nd] [Lbl] [C] [STO] 4 [2nd] [Prt] [R/S]	Store R_2 in memory 04.
024	[2nd] [Lbl] [2nd] [C'] [RCL] 4 [÷] [RCL] 3 [=] [2nd] [Prt] [R/S]	Calculate and display R_1.

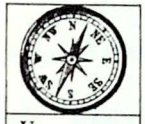


Step #	Program Keystrokes	Purpose/Comments
034	2nd Lbl D 1 ÷ (2 × 2nd π × RCL 1 × RCL 4 ÷ RCL 3) = 2nd Prt R/S	Calculate and display C_1.
055	2nd Lbl 2nd D' 1 ÷ (2 × 2nd π × RCL 2 × RCL 4 = 2nd Prt R/S	Calculate and display C_2.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}
R_{01} f_1
R_{02} f_2
R_{03} G
R_{04} R_2



USING THE PROGRAM

Procedure	Press	Display/Comments
Enter f_1.	A	f_1°
Enter f_2.	2nd A'	f_2°
Enter G.	B	G°
Enter R_2.	C	R_2°
Clear display.	CLR	0
Place calculator in engineering mode.	2nd Eng	0. 00
Fix decimal to desired accuracy.	2nd Fix n	In most situations, n = 2 is desired accuracy.
Calculate the value of R_1.	2nd C'	R_1°
Calculate the value of C_1.	D	C_1° in farads.
Calculate the value of C_2.	2nd D'	C_2° in farads.

°These values are printed if the PC-100A is used.

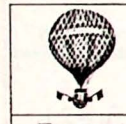
9

Active Filter to Improve Signal-to-Noise Ratio

Sample Problem

Given the parameters $R_2 = 10{,}000$ ohms, $G = 1$, $f_1 = 0.002$ Hz and $f_2 = 0.5$ Hz, find the values of R_1 (ohms), C_1 (farads), and C_2 (farads) for the band-pass filter in *Figure 1.*

Procedure	Press	Display/Comments
Enter program carefully.		0
Enter f_1.	.002 [A]	0.002
Enter f_2.	.5 [2nd] [A']	0.5
Enter G.	1 [B]	1.
Enter R_2.	10 [EE] 3 [C]	1. 04
Clear display.	[CLR]	0
Place calculator in engineering mode.	[2nd] [Eng]	0. 00
Fix the decimal point for desired accuracy.	[2nd] [fix] 2	0.00 00
Press [2nd] [C'] for value of R_1.	[2nd] [C']	10.00 03 This is R_1.
Press [D] for value of C_1.	[D]	7.96-03 This is C_1.
Press [2nd] [D'] for value of C_2.	[2nd] [D']	31.83-06 This is C_2.



Further Explorations

1. Design a high-pass filter for which $G = 1$, and $f_1 = 0.002$ Hz. Repeat for a low-pass filter with $G = 1$, $f_2 = 0.5$ Hz.
2. Determine appropriate component values for the filters in *Figures 3* and *4* given in each case a passband gain G, a cutoff frequency f_0, damping ratio ζ and capacitance C.

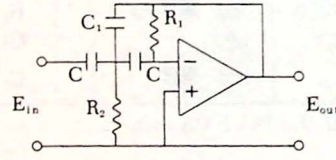


Figure 3. High-Pass Filter

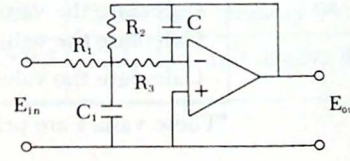
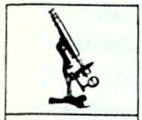


Figure 4. Low-Pass Filter

References/ Information

J. G. Graeme and G. E. Tobey, *Operational Amplifiers: Design and Application* (New York: McGraw-Hill, Inc., 1971).

Estimating Number of Samples for Specified Standard Error



OBJECTIVE

Due to noise in the passband, sampled values of transducer output can show variation even after filtering, but averaging of multiple measurements can reduce error. This program allows you to specify the standard error (e) and the percent confidence desired. The program then computes the number of samples necessary for the measurement average to estimate the population mean. Traditionally calculations of this kind have required referencing of statistical tables. However, the problem can be solved on the calculator, without tables, using two Master Library programs.

FACTS/ THEORY

Additive Gaussian white noise is assumed so that the noise associated with each measurement is an independent variable. As the number of measurements (N) increases, the mean of the measurements ($\bar{x}$) approaches the true mean (μ), which would be obtained if there was no noise. The value of N must be determined such that

$$\text{Probability} \left\{ |\bar{x} - \mu| < e \right\} = \alpha \qquad [1]$$

The magnitude of the difference $\bar{x} - \mu$ is called the error of estimate. In the above relationship 100α is described as the percent confidence that the error of estimate is less than e. Even though μ is unknown, one can still speak of how close $\bar{x}$ is to μ, and therefore can find the number of samples necessary to obtain the desired accuracy of the estimate of μ.

The equation to find N is given by

$$N = \left[\frac{x_{\alpha/2}\sigma}{e} \right]^2 \qquad [2]$$

where σ is the standard deviation of the samples,* e is the allowable error of estimate, α is the probability found in equation [1], and $x_{\alpha/2}$ is the number of standard deviations of a Gaussian variable x such that the

$$\text{Probability} \left\{ x < x_{\alpha/2} \right\} = \alpha/2.$$

To find $x_{\alpha/2}$ for a given α, one needs to solve the following equation for $x_{\alpha/2}$.

$$\frac{1-\alpha}{2} = f(x_{\alpha/2}) \qquad [3]$$

We can use the calculator to solve for $x_{\alpha/2}$ by rewriting equation [3] as

$$g(x) = f(x_{\alpha/2}) - \frac{1-\alpha}{2} = 0 \qquad [4]$$

*If the standard deviation is unknown, it can be estimated by calculating σ for a preliminary sample using the statistics functions described on page V-32 of *Personal Programming*.

9

Estimating Number of Samples for Specified Standard Error

The function f that we are using is the area function Q(x) determined by Master Library Program ML-14. This particular function was chosen since a Gaussian or normal distribution of the means was assumed. Once equation [4] is generated, we will use Master Library program ML-8 to find the $x_{\alpha/2}$ value that makes g(x) equal 0. Then we can evaluate equation [2] for the number of samples we must use.

The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [A] [÷] 100 [+/−] [+] 1 [=] [÷] 2 [=] [STO] 10 [INV] [SBR]	Enter percent confidence (α). Calculate and store $\frac{1-(\alpha/100)}{2}$.
016	[2nd] [Lbl] [B] [STO] 21 [2nd] [Prt] [INV] [SBR]	Enter and store σ.
022	[2nd] [Lbl] [E] [STO] 22 [2nd] [Prt] [INV] [SBR]	Enter and store error e.
028	[2nd] [Lbl] [C] [(] [D] [×] [RCL] 21 [÷] [RCL] 22 [)] [x^2] [2nd] [fix] 0 [2nd] [Prt] [INV] [SBR]	Calculate and display sample size N. $N = \left[\frac{x_{\alpha/2}\sigma}{e}\right]^2$
044	[2nd] [Lbl] [2nd] [A'] [(] [2nd] [Pgm] 14 [A] [2nd] [Pgm] 14 [B] [−] [RCL] 10 [)] [INV] [SBR]	Generate function to be zeroed. $f(x_{\alpha/2}) - \frac{(1-\alpha)}{2} = 0$
058	[2nd] [Lbl] [D] 15.11 [+/−] [2nd] [Pgm] 08 [A] [+/−] [2nd] [Pgm] 08 [B] [2nd] [Pgm] 08 [E] [INV] [SBR]	Initialize program ML-08 to find the root of the function generated in [SBR] [A']. Note that −15.11 and 15.11 are the maximum limits for ML-08.
	[LRN] [RST]	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

Estimating Number of Samples for Specified Standard Error

Register Contents				
	$R_{01}-R_{09}$	ML-08	R_{21}	σ
	$R_{01}-R_{03}$	ML-14	R_{22}	e
	R_{10}	$[1-(\alpha/100)]/2$		

Using the Program

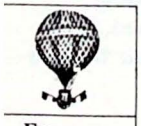
Procedure	Press	Display/Comments
Enter percent confidence.	A	$\frac{1-(\alpha/100)}{2}$
Enter σ.	B	σ°
Enter allowable error estimate.	E	e°
Calculate sample size.	C	N°

°This value is printed if the PC-100A is used.

Sample Problem

A preliminary sample of cardiac output yielded a standard deviation (σ) of 0.3. How large a sample is required if we want to be 95% confident that our estimate of the true mean μ is off by less than 0.05?

Procedure	Press	Display/Comments
Enter program carefully.		0
Return calculator to standard display.	INV 2nd fix	0.
Enter percent confidence.	95 A	0.025
Enter standard deviation (σ).	.3 B	0.3
Enter error e.	.05 E	0.05
Press C for the sample size (N).	C	138.



Further Explorations

1. Write a program that will compute the percent confidence for a given sample size, standard deviation, and error.
2. When using Master Library Program ML-08, the value for the maximum error E is automatically set to 0.01. Investigate the effects of changing E on sample size determination.

References/ Information

R. E. Walpole and R. H. Meyers, *Probability and Statistics for Engineers and Scientists* (New York: Macmillan Publishing Co., Inc., 1978).

9 Transducer Scaling and Interactions

OBJECTIVE

The outputs of transducers are typically voltage levels which actuate meters or other readout devices. Additional methods can be employed to linearize the transducer output, make zero or baseline adjustment, and perform scaling of measurements. When sample values are read sequentially, a residue of earlier samples may remain in the transducer and affect readings. Recognition of this last problem and correction or compensation for it may be necessary. The program below solves the linear segment of this problem by finding the interaction matrix needed to correct readings and provide scaling.

FACTS/ THEORY

The matrix equation relating corrected readings to sampled values of transducer output voltages is:

$$\hat{P}_i = \hat{B}_0 + K(P_i - B_0) \qquad [1]$$

where $\hat{P}_i$ = an Nx1 column matrix representing true (corrected) gas tensions for N sequential samples.

P_i = an Nx1 column matrix of transducer output voltages to be corrected.

$\hat{B}_0$ = an Nx1 column matrix of true gas tensions for a set of N sequential samples.

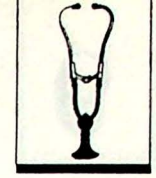
B_0 = an Nx1 column matrix of transducer output voltages from the set of N sequential samples.

K = an NxN matrix representing a correction constant.

Note: Matrix elements are identified by italicized symbols.

To find $\hat{B}_0$, N sequential samples of calibrated gases having true gas tensions, $\hat{B}_1, \ldots, \hat{B}_N$, are passed through the transducer. The transducer outputs, $B_1, \ldots, B_N$, from the N sequential samples give the values for the column matrix B_0. The two column matrices $\hat{B}_0$ and B_0 are called baselines.

To determine K for a particular system, calibrated gases with known gas tensions are used in the system. In addition to finding baseline values, N more trials with N sequential samples of the calibrated gases are used. The N additional trials have known gas tensions $\hat{P}_1, \ldots, \hat{P}_N$ and corresponding transducer voltage readings $P_1, \ldots, P_N$. When K has been determined, it can then be applied to the transducer outputs for other gases to find their true gas tensions.



Consider for example N = 3. Then we have

$$\hat{P}_i = \begin{bmatrix} \hat{P}_{1i} \\ \hat{P}_{2i} \\ \hat{P}_{3i} \end{bmatrix} \quad P_i = \begin{bmatrix} P_{1i} \\ P_{2i} \\ P_{3i} \end{bmatrix} \quad \hat{B}_o = \begin{bmatrix} \hat{B}_1 \\ \hat{B}_2 \\ \hat{B}_3 \end{bmatrix} \quad B_o = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}$$

Once the calibration data is collected, it is used in the following matrix equation.

$$\hat{P} = \hat{B} + K(P - B) \qquad [2]$$

where $\hat{P} = [\hat{P}_1 \quad \hat{P}_2 \quad \hat{P}_3]$ $\qquad \hat{B} = [\hat{B}_o \quad \hat{B}_o \quad \hat{B}_o]$

$P = [P_1 \quad P_2 \quad P_3]$ $\qquad B = [B_o \quad B_o \quad B_o]$

We can rewrite equation [2] as:

$$\hat{P} - \hat{B} = K(P - B) \qquad [3]$$

Now if we define

$$\hat{P}' = \hat{P} - \hat{B} \text{ and } P' = P - B$$

then equation [3] becomes:

$$\hat{P}' = K\,P' \qquad [4]$$

Solving equation[4] for K yields:

$$K = \hat{P}'P'^{-1} \qquad [5]$$

In solving for K, two Master Library programs will be used. To find P'^{-1} (inverse matrix of P′), we simply use program ML-02. Matrix multiplication can be done quickly using program ML-03. Remember that matrix multiplication is not commutative ($AB \neq BA$), so be sure you perform the multiplication in the order shown. Also, note that $\hat{P}'$ and P'^{-1} must both be nonzero matrices to insure that K is nonzero. This can be assured by the choice of calibration gases used. In addition, P′ must have a nonzero determinant in order to compute P'^{-1}.

Note that you can choose the calibration gas sequences so that $\hat{P}'$ is a diagonal matrix with diagonal elements α ($\alpha \neq 0$). This insures that $\hat{P}'$ is nonzero and also simplifies equation [5]. Thus we have the following equation:

$$K = \alpha P'^{-1}$$

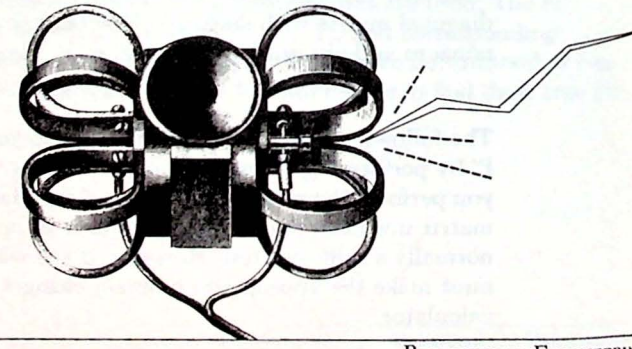
The following program is designed to help you compute the matrices $\hat{P}'$ and P′ by performing the matrix subtractions for you. Also, the program will help you perform the matrix inversion and calculate $K = \alpha P'^{-1}$. Note that the matrix inversion part of the program is set up for a 3 x 3 matrix. This is normally a sufficient test. However, if you wish to use a larger matrix, you must make the appropriate program changes and may have to repartition the calculator.

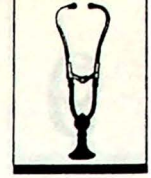
THE PROGRAM

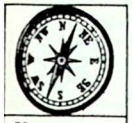
Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [A] 8 [STO] 0 [CLR] [R/S]	Initializes memory 0. This register is used for indirect accessing of data.
007	[2nd] [Lbl] [B] [STO] [2nd] [Ind] 0 [2nd] [Op] 20 [R/S]	Subroutine to load the first matrix (P) into memory.
014	[2nd] [Lbl] [C] [INV] [SUM] [2nd] [Ind] 0 [RCL] [2nd] [Ind] 0 [2nd] [Op] 20 [R/S]	Subroutine to subtract the second matrix (B) from the first matrix (P) giving matrix P′.
024	[2nd] [Lbl] [D] 3 [2nd] [Pgm] 02 [A] [CLR] [2nd] [Pgm] 02 [2nd] [E^x] [R/S]	Calls Master Library Program 02 to calculate P'^{-1}.
035	[2nd] [Lbl] [E] [RCL] 1 [2nd] [Prd] [2nd] [Ind] 0 [RCL] [2nd] [Ind] 0 [2nd] [Op] 20 [R/S]	If $\hat{P}'$ is a constant diagonal matrix, this subroutine may be used to compute $\alpha P'^{-1}$. α must be stored in memory 1 before running this subroutine.
	[LRN] [RST]	Exits learn mode.

REGISTER CONTENTS

R_{00} Used
R_{01} α
$R_{02} - R_{07}$ Used by ML-02
$R_{08} - R_{22}$ P, P′, P'^{-1}







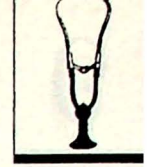
USING THE PROGRAM

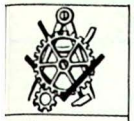
Procedure	Press	Display/Comments	
Clear memories.	CLR 2nd CMS	0.	
Initialize program.	A	0	
Enter P matrix by columns.			
P_{11}	B	P_{11}	
P_{21}	B	P_{21}	
.	.	.	
.	.	.	
.	.	.	
P_{33}	B	P_{33}	
Reset program.	A	0	
Enter B matrix by columns. Note that the columns are identical.			
B_1	C	P'_{11}	$P_{11}-B_1$
B_2	C	P'_{21}	$P_{21}-B_2$
B_3	C	P'_{31}	$P_{31}-B_3$
B_1	C	P'_{12}	$P_{12}-B_1$
.	.	.	
.	.	.	
.	.	.	
B_3	C	P'_{33}	$P_{33}-B_3$
Calculate the P'^{-1} matrix.	D	determinant*	This is the determinant of the P′ matrix.
Fix decimal point to desired accuracy (n places).	2nd Fix n		
Calculate K (Method 1). (Use only if $\hat{P}'$ is a constant diagonal matrix.)			
Initialize memory 0.	A	0	
Store α in memory 1.	α STO 01	α	

*This value is printed if the PC-100A is used.

Procedure	Press	Display /Comments
Calculate		
K_{11}	[E]	K_{11}
K_{21}	[E]	K_{21}
.	.	.
.	.	.
.	.	.
K_{33}	[E]	K_{33}
OR		
Calculate K (Method 2). (Use if $\hat{P}'$ is not a constant diagonal matrix.)		
To use program ML-03 you must move the P'^{-1} matrix already calculated to memories 30-38.	[2nd] [Pgm] 02	
	1 [2nd] [C']	
	[R/S] [STO] 30	P'^{-1}_{11} *
	[R/S] [STO] 31	P'^{-1}_{21} *
	.	.
	.	.
	.	.
	[R/S] [STO] 38	P'^{-1}_{33} *
Exit Program ML-02.	[RST]	
See *Master Library*, pp. 14-15 for directions on matrix multiplication.		

*This value is printed if the PC-100A is used.





SAMPLE PROBLEM

Consider the use of two calibration gases; one contains oxygen at a partial pressure of 50 mm Hg and the other contains oxygen at a partial pressure of 147 mm Hg. The lower pressure gas is used to obtain the baseline values and three sequential samples of this gas produce transducer outputs of 75.6 mV, 100 mV, and 109 mV. This gives us the following baseline matrices:

$$\hat{B}_o = \begin{bmatrix} 50 \\ 50 \\ 50 \end{bmatrix} \qquad B_o = \begin{bmatrix} 75.6 \\ 100 \\ 109 \end{bmatrix}$$

For the rest of the test, the gas samples used and transducer outputs are shown in the tables below.

Sequential Sample Number	*Calibration Gas (mm Hg)*			*Transducer Output (mV)*		
	$\hat{P}_1$	$\hat{P}_2$	$\hat{P}_3$	P_1	P_2	P_3
1	147	50	50	222	75.6	75.6
2	50	147	50	144	246	100
3	50	50	147	128	153	255

Procedure	Press	Display/Comments
Enter program.		0
Clear data registers.	CLR 2nd CMs	0.
Initialize program.	A	0
Enter P matrix.		
P_{11}	222 B	222.
P_{21}	144 B	144.
P_{31}	128 B	128.
P_{12}	75.6 B	75.6
P_{22}	246 B	246.
P_{32}	153 B	153.
P_{13}	75.6 B	75.6
P_{23}	100 B	100.
P_{33}	255 B	255.
Reset program.	A	0

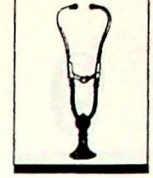
Procedure	Press	Display/Comments
Enter B matrix.		
B_1	75.6 C	146.4
B_2	100 C	44.
B_3	109 C	19.
B_1	75.6 C	0.
B_2	100 C	146.
B_3	109 C	44.
B_1	75.6 C	0.
B_2	100 C	0.
B_3	109 C	146.
Calculate the P'^{-1} matrix.	D	3120662.4 Wait for display value before continuing.
Fix decimal point. (We'll use 3 for this example.)	CLR 2nd fix 3	0.000

To calculate the $\hat{P}'$ matrix we must subtract $\hat{B}$ from $\hat{P}$. This is easily done in our example since $\hat{B}$ is the matrix consisting of three columns of the $\hat{B}_o$ matrix. Thus $\hat{B}$ is a matrix with the value 50 in every position. Thus our $\hat{P}'$ matrix can be computed easily as follows.

$$\hat{P}' = \hat{P} - \hat{B} = \begin{bmatrix} 97 & 0 & 0 \\ 0 & 97 & 0 \\ 0 & 0 & 97 \end{bmatrix}$$

Since this is a constant diagonal matrix we will use method 1 to calculate K.

Procedure	Press	Display/Comments
Initialize memory 0.	A	0
Store α in memory 1.	97 STO 01	97.000
Calculate K.		
K_{11}	E	0.663
K_{21}	E	−0.200
K_{31}	E	−0.026
K_{12}	E	0.000
K_{22}	E	0.664
K_{32}	E	−0.200
K_{13}	E	0.000
K_{23}	E	0.000
K_{33}	E	0.664



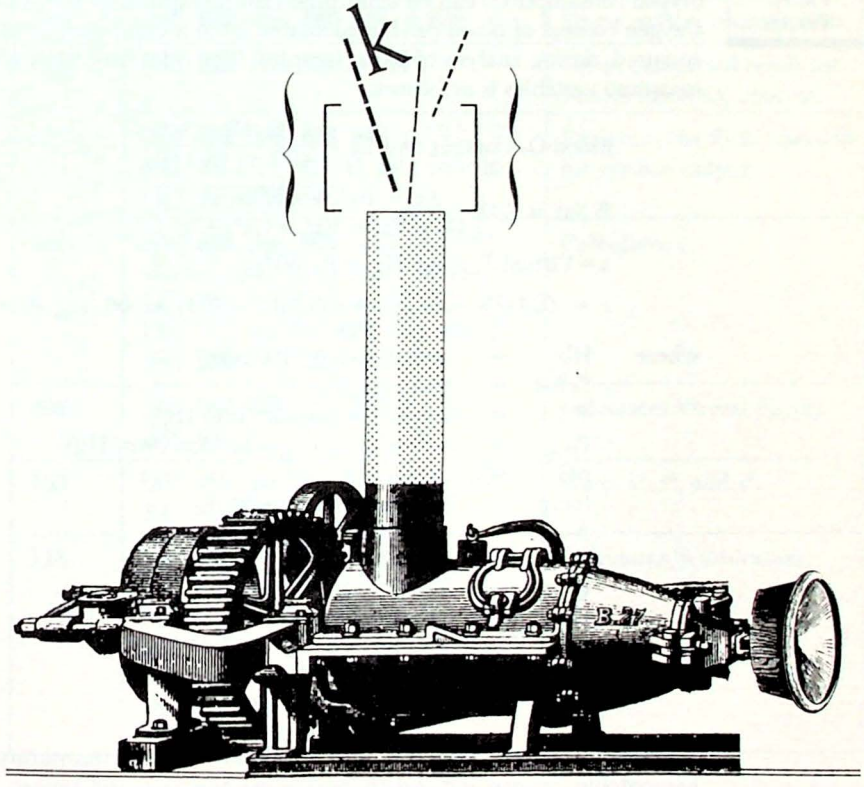
Thus our K matrix is:

$$K = \begin{bmatrix} 0.663 & 0.000 & 0.000 \\ -0.200 & 0.664 & 0.000 \\ -0.026 & -0.200 & 0.664 \end{bmatrix}$$

This matrix can now be used in equation 1 to convert the transducer outputs of unknown gases into their true gas tensions.

REFERENCES / INFORMATION

Bernard Kolman, *Introductory Linear Algebra with Applications* (New York: Macmillan Publishing Co., Inc., 1976).



9 Cardiac Output Using the Fick Equation

OBJECTIVE

Cardiac output is calculated by entering the appropriate data into the Fick equation:

$$\text{Cardiac Output} = \frac{\text{Oxygen Consumption}}{\text{Arterial Blood } O_2 \text{ Content} - \text{Venous Blood } O_2 \text{ Content}}$$

Oxygen contents of blood may in turn be calculated from oxygen and carbon dioxide gas tensions, hemoglobin level, temperature, and pH.

FACTS/ THEORY

The Fick method for measuring cardiac output is based on simple mass balance. The rate of blood flow through the heart and lungs can be determined by measuring the uptake of oxygen and the change in concentration of oxygen as blood passes through. The oxygen uptake (or oxygen consumption) can be determined from respiratory gas measurements. Oxygen content of blood can be calculated from a combination of data acquired during analysis of blood samples. The relationship of content to measured variables is as follows.

$$\text{Blood } O_2 \text{ Content (vol \%)} = \frac{1.39\,(\text{Hb})\,(\%\ \text{Sat})}{100 + .003\,(P_{O_2})}$$

$$\%\ \text{Sat} = 100\left[\frac{Ax + Bx^2 + Cx^3 + x^4}{D + Ex + Fx^2 + Gx^3 + x^4}\right]$$

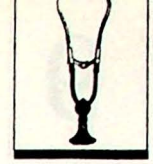
$$x = \text{Virtual } P_{O_2}\,(\text{mm Hg}) = P_{O_2}\,10^{Z}$$

$$z = .024\,(37 - \text{temp}) + .40\,(\text{pH} - 7.4) + .06\,(\log 40 - \log P_{CO_2})$$

where

Hb	=	hemoglobin (g/100 ml),
% Sat	=	% saturation,
P_{O_2}	=	oxygen gas tension (mm Hg),
P_{CO_2}	=	carbon dioxide gas tension (mm Hg),
pH	=	blood pH,
temp	=	blood temperature (°C),
A	=	−8532.2289,
B	=	2121.4010,
C	=	−67.073989,
D	=	935960.87,
E	=	−31346.258,
F	=	2396.1674, and
G	=	67.104406

The program which follows is written so that values of temperature, pH, hemoglobin, oxygen and carbon dioxide gas tensions, and oxygen consumption are entered under separate user labels. Any entry can be changed without affecting other previously stored values.



The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 1 R/S	Stores O_2 gas tension.
005	2nd Lbl B STO 2 R/S	Stores CO_2 gas tension.
010	2nd Lbl C STO 3 R/S	Stores blood pH.
015	2nd Lbl D STO 4 R/S	Stores blood hemoglobin.
020	2nd Lbl E STO 5 R/S	Stores blood temperature.
025	2nd Lbl 2nd E' STO 6 R/S	Stores oxygen consumption.
030	2nd Lbl EE STO 11 R/S	Stores calculated result for venous blood O_2 content.
035	2nd Lbl 2nd C' RCL 6 ÷ (RCL 10 − RCL 11 = ÷ 10 = 2nd Prt R/S	Evaluates the Fick equation for cardiac output.
053	2nd Lbl 2nd D' .024 × (37 − RCL 5) + .4 × (RCL 3 − 7.4) + .06 × (40 2nd log − RCL 2 2nd log = STO 07	Calculates z.
095	INV 2nd log × RCL 1 = STO 08	Calculates Virtual P_{O_2} (x).
103	x^2 STO 20 × RCL 8 = STO 21 × RCL 8 = STO 22	Stores x^2, x^3, and x^4.
118	RCL 13 × RCL 8 + RCL 14 × RCL 20 + RCL 15 × RCL 21 + RCL 22 = ÷ (RCL 16 + RCL 17 × RCL 8 + RCL 18 × RCL 20 + RCL 19 × RCL 21 + RCL 22 = × 100 = STO 09	Calculates % saturation.

Cardiac Output Using the Fick Equation

Step #	Program Keystrokes	Purpose/Comments
172	1.39 [×] [RCL] 4 [×] [RCL] 9 [=] [÷] [(] 100 [+] .003 [×] [RCL] 1 [=] [INV] [2nd] [If flg] 1 [EE] [STO] 10 [R/S]	Calculates blood O_2 content. If venous blood O_2 content has been calculated (flag 1 not set), transfers to subroutine [EE] and stores Cv_{O_2} in memory 11. If arterial blood O_2 content has been calculated (flag 1 set), stores Ca_{O_2} in memory 10.
204	[2nd] [Lbl] [2nd] [A'] [2nd] [St flg] 1 [2nd] [D']	Starts calculation of Ca_{O_2} (arterial blood O_2 content).
209	[2nd] [Lbl] [2nd] [B'] [INV] [2nd] [St flg] 1 [2nd] [D']	Starts calculation of Cv_{O_2} (venous blood O_2 content).
	[LRN] [RST]	Exits learn mode.

Note: Print command may be omitted if the PC-100A is not used.

Register Contents

R_{00}		R_{08}	Virtual P_{O_2}	R_{16}	D
R_{01}	P_{O_2}	R_{09}	% Sat	R_{17}	E
R_{02}	P_{CO_2}	R_{10}	Ca_{O_2}	R_{18}	F
R_{03}	pH	R_{11}	Cv_{O_2}	R_{19}	G
R_{04}	Hb	R_{12}		R_{20}	x^2
R_{05}	temp	R_{13}	A	R_{21}	x^3
R_{06}	O_2 consumption	R_{14}	B	R_{22}	x^4
R_{07}	z	R_{15}	C	R_{23}	

Note: Intermediate results are stored during calculation and can be recalled from memories if needed.

Using the Program

Procedure	Press	Display/Comments
Set decimal for desired accuracy.	2nd Fix n	(Optional)
Enter constants.		
A	STO 13	A
B	STO 14	B
C	STO 15	C
D	STO 16	D
E	STO 17	E
F	STO 18	F
G	STO 19	G
Enter arterial blood data.		
P_{O_2}	A	P_{O_2}
P_{CO_2}	B	P_{CO_2}
pH	C	pH
Hb	D	Hb
temperature	E	temp.
oxygen consumption	2nd E'	O_2 consumption
Calculate arterial oxygen content.	2nd A'	Ca_{O_2}
Enter venous blood data.		
P_{O_2}	A	P_{O_2}
P_{CO_2}	B	P_{CO_2}
pH	C	pH
Calculate venous blood oxygen content.	2nd B'	Cv_{O_2}
Calculate cardiac output using the Fick equation.	2nd C'	C.O.*

*This value is printed if the PC-100A is used.

9

Cardiac Output Using the Fick Equation

SAMPLE PROBLEM

Find the cardiac output given the following information.

Arterial Blood Data
$Pa_{O_2} = 123$ mm Hg
$Pa_{CO_2} = 31.8$ mm Hg
$pH_a = 7.47$

Venous Blood Data
$Pv_{O_2} = 40.3$ mm Hg
$Pv_{CO_2} = 38.1$ mm Hg
$pH_v = 7.45$

Hb = 16 g/100 ml
temp = 37.1°C
O_2 consumption = 317 ml/min

Procedure	Press	Display/Comments
Enter program. Set decimal for desired accuracy.	2nd fix 2	(We'll use a 2-place decimal setting for convenience.)
Enter constants.		
A	8532.2289 +/− STO 13	−8532.23
B	2121.401 STO 14	2121.40
C	67.073989 +/− STO 15	−67.07
D	935960.87 STO 16	935960.87
E	31346.258 +/− STO 17	−31346.26
F	2396.1674 STO 18	2396.17
G	67.104406 +/− STO 19	−67.10
Enter arterial blood data.		
Pa_{O_2}	123 A	123.00
Pa_{CO_2}	31.8 B	31.80
pH_a	7.47 C	7.47
Enter hemoglobin (Hb) in g/100 ml.	16 D	16.00
Enter temperature in °C.	37.1 E	37.10
Enter oxygen consumption in ml/min.	317 2nd E'	317.00
Calculate arterial oxygen content.	2nd A'	21.85

Procedure	Press	Display/Comments
Enter venous blood data.		
Pv_{O_2}	40.3 [A]	40.30
Pv_{CO_2}	38.1 [B]	38.10
pH_v	7.45 [C]	7.45
Calculate venous oxygen content.	[2nd] [B']	17.09
Calculate cardiac output.	[2nd] [C']	6.67 litres/min

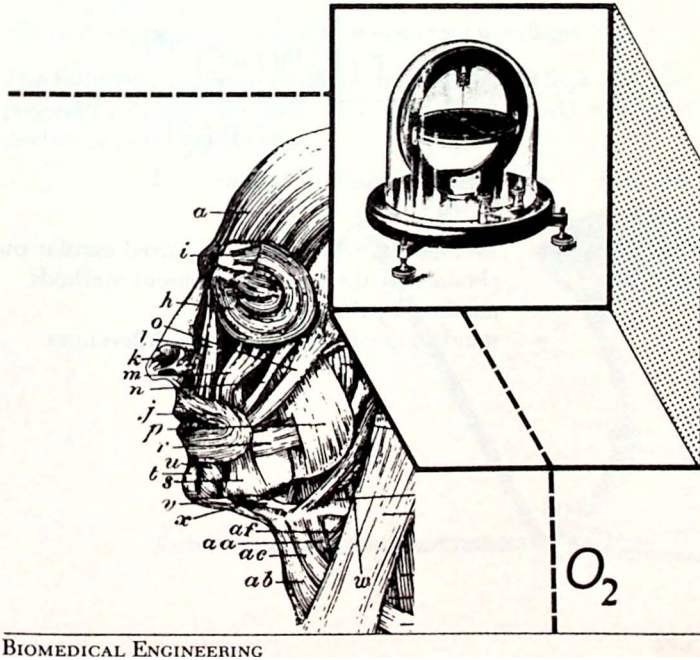
For a P_{O_2} less than 10 mm Hg, the equation for % saturation should be changed to

$$\%Sat = 100\ (.003683x + .000584x^2).$$

Revise the program to allow for this situation.

REFERENCES/ INFORMATION

J. B. West and P. D. Wagner, "Pulmonary Gas Exchange" in *Bioengineering Aspects of the Lung*, ed. by J. B. West (New York: Marcel Dekker, 1977).



9 Student's t-Test

OBJECTIVE

Once the design and implementation of a new measurement system are complete, the system must be evaluated. One approach to the evaluation is to match the results from the new system with data from a reference measurement system. When true cardiac output becomes constant, data can be collected in pairs that contain one cardiac output determination from each method. For example, blood gas samples for the Fick method might be collected from a patient at the same time that a dye dilution test is performed. Both the Fick method and the dye system should give you essentially the same results but with some measurement errors in each. The calculator is employed to perform the student's t-test on the data. From the results of the test you can determine the likelihood that any differences in results from the two methods are due to measurement errors and that the average difference of repeated cardiac measurements from the two methods would approach zero.

FACTS/ THEORY

The student's t-distribution is used here under the assumption that the data is drawn from a Gaussian distribution and that the variance of the data must be estimated. The calculator is used to compute the probability that an observed average difference (or any lesser difference) in paired data could have occurred by chance, assuming that the true mean difference is zero.

To find the desired probability, the following integral must be evaluated.

$$h(t,\nu) = \int_{-t}^{t} \frac{\Gamma\left(\frac{\nu+1}{2}\right)\left(1+\frac{y^2}{\nu}\right)^{-\left(\frac{\nu+1}{2}\right)}}{\sqrt{\pi\nu}\,\Gamma\left(\frac{\nu}{2}\right)}\,dy$$

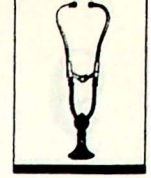
where: ν = degrees of freedom, usually $n-1$

$t = \dfrac{\overline{\Delta x}\sqrt{n}}{s}$

$\overline{\Delta x}$ = average difference between paired cardiac output values obtained by the two measurement methods

n = number of paired samples

s = standard deviation estimate of differences



The desired probability is then given by

$$P = 1 - h(t,\nu)$$

If P is greater than 0.5, then the average difference between the two methods is insignificant. However, if $P < 0.5$, then there is a significant difference between the two methods, and you should not assume that the differences are due to measurement error.

The integral given above may be easily evaluated using series approximations. There are three difference approximations, depending on the value of ν.

ν even

$$h(t,\nu) = \sin\theta\left(1 + \frac{1}{2}\cos^2\theta + \frac{1\cdot3}{2\cdot4}\cos^4\theta + \ldots + \frac{1\cdot3\cdot5\ldots(\nu-3)}{2\cdot4\cdot6\ldots(\nu-2)}\cos^{(\nu-2)}\theta\right)$$

$\nu = 1$

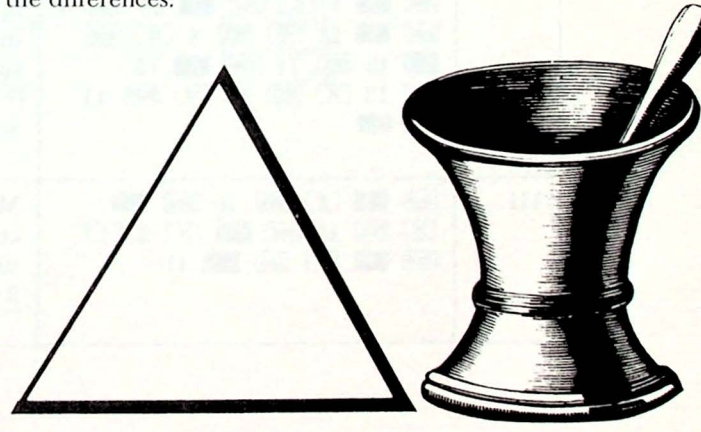
$$h(t,\nu) = \frac{2\theta}{\pi}$$

ν odd, $\nu \neq 1$

$$h(t,\nu) = \frac{2\theta}{\pi} + \frac{2}{\pi}\cos\theta\sin\theta\left(1 + \frac{2}{3}\cos^2\theta + \ldots + \frac{2\cdot4\ldots(\nu-3)}{3\cdot5\ldots(\nu-2)}\cos^{(\nu-3)}\theta\right)$$

where $\theta = \arctan(t/\sqrt{\nu})$ with θ expressed in radians.

The following program evaluates these series and finds the value P. The program will also calculate t, the mean difference $\overline{\Delta x}$, and the standard deviation of the differences.



9 Student's t-Test

THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A = 2nd $\Sigma+$ R/S	Assimilates the difference of each sample.
005	2nd Lbl 2nd A' = INV 2nd $\Sigma+$ R/S	Deletes faulty sample (if necessary).
011	2nd Lbl B RCL 3 $\sqrt{x}$ × 2nd $\bar{x}$ ÷ INV 2nd $\bar{x}$ = INV SBR	Computes t. $\frac{t=\overline{\Delta x}\sqrt{n}}{s}$
023	2nd Lbl D 2nd Rad B ÷ (RCL 3 − 1) STO 7 $\sqrt{x}$ = INV 2nd tan STO 10	Calculates $\theta = \arctan(t/\sqrt{\nu})$ and stores it in memory 10. Note this automatically computes $\nu = n - 1$.
042	1 STO 8 STO 13 STO 15 0 STO 11 RCL 10 2nd cos x^2 STO 14	Initializes registers for calculation of P.
058	RCL 7 ÷ 2 = STO 0 $x \geq t$ RCL 0 2nd Int 2nd $x=1$ E INV 2nd Dsz 7 2nd B 1 STO 11	Determines status of ν. If $\nu = 1$, transfers to subroutine B. If ν is even, transfers to subroutine E.
078	2nd Lbl 2nd E' 2nd Op 37 INV 2nd Dsz 7 C 2nd Op 28 RCL 8 2nd Op 28 ÷ RCL 8 = 2nd Fix 13 RCL 14 2nd Prd 15 RCL 13 × RCL 15 = SUM 11 2nd E'	For ν odd, $\nu \neq 1$, computes $h(t,\nu)$. This sum is stored in memory 11. When the summation reaches the $(\nu - 3)$ term, transfers to subroutine C.
111	2nd Lbl C RCL 10 2nd cos × RCL 10 2nd sin × 2 ÷ 2nd π = 2nd Prd 11	Multiplies the sum computed in subroutine E' by $\frac{2\sin\theta\cos\theta}{\pi}$.

Step #	Program Keystrokes	Purpose/Comments
127	2nd Lbl 2nd B' RCL 10 × 2 ÷ 2nd π = SUM 11 RCL 11 +/− + 1 = R/S	Evaluates $\frac{2\theta}{\pi}$ and adds it to what is in memory 11. (Note this is a zero if $\nu = 1$, and is the value computed by E' and C if ν is odd and $\nu \neq 1$.) Then computes $P = 1 - h(t,\nu)$.
145	2nd Lbl E 1 STO 11 2nd Lbl 2nd C' 2nd Op 37 INV 2nd Dsz 7 2nd D' RCL 8 2nd Op 28 ÷ RCL 8 2nd Op 28 = 2nd Prd 13 RCL 14 2nd Prd 15 RCL 13 × RCL 15 = SUM 11 2nd C'	For ν even, computes $h(t,\nu)$. This sum is stored in memory 11. When the summation reaches the $(\nu-2)$ term, transfers to subroutine D'.
183	2nd Lbl 2nd D' RCL 10 2nd sin 2nd Prd 11 RCL 11 +/− + 1 = R/S	Multiplies the sum computed in subroutine E by $\sin\theta$. Then computes $P = 1 - h(t,\nu)$.
	LRN RST	Exits learn mode.

REGISTER CONTENTS

$R_{01} - R_{06}$	Statistics	R_{11}	$h(t,\nu)$
R_{07}	Counter	R_{12}	Used
R_{08}	Used	R_{13}	Used
R_{09}		R_{14}	$\cos^2\theta$
R_{10}	Used	R_{15}	$\cos^{2n}\theta$

9 Student's t–Test

Using the Program

Procedure	Press	Display/Comments
Clear the memories.	CLR 2nd CMs	0.
Fix the decimal point to desired accuracy (n places).	2nd Fix n	
Enter data pairs.		
First value	−	First value
Second value	A	Pair number
Repeat for each data pair.		
Remove faulty data pair.		
First value (as input).	−	First value
Second value (as input).	2nd A'	Previous pair number
Calculate P.	D	P
Optional:		
Calculate t.	B	t
Calculate mean difference.	2nd $\bar{x}$	$\Delta\bar{x}$
Calculate standard deviation of differences.	INV 2nd $\bar{x}$	s

Note: If the information stored in data registers 01-06 by pressing A is already known, it can be entered directly into those registers as shown in *Personal Programming*, p. V-33.

Sample Problem

Suppose the following data were collected from two different systems. Use the program to calculate $\Delta\bar{x}$, s, and P. Is there a significant difference between the two systems?

System 1	*System 2*
6.33	6.14
6.67	6.85
5.42	5.50
7.96	7.43
9.66	8.49
4.42	5.12

Procedure	Press	Display/Comments
Enter program.		0
Clear data registers.	CLR 2nd CMs	0.
Select 3-decimal place display.	2nd fix 3	0.000
Enter data pairs.	6.33 −	6.330
	6.14 A	1.000
	6.67 −	6.670
	6.85 A	2.000
	5.42 −	5.420
	5.5 A	3.000
	7.96 −	7.960
	7.43 A	4.000
	9.66 −	9.660
	8.49 A	5.000
	4.42 −	4.420
	5.12 A	6.000
Calculate mean difference.	2nd $\bar{x}$	0.155 $\overline{\Delta x}$
Calculate standard deviation of difference.	INV 2nd $\bar{x}$	0.644 s
Calculate P.	D	0.581 P

The average difference between the two methods is insignificant.

REFERENCES/ INFORMATION

U.S. Department of Commerce, National Bureau of Standards, *Handbook of Mathematical Functions*, M. Abramowitz and Irene Stegun, eds. (Washington, D.C.: U.S. Government Printing Office, 1968).

R. E. Walpole and R. H. Myers, *Probability and Statistics for Engineers* (New York: Macmillan Publishing Co., Inc., 1978).

Personal Programming, pp. V-32ff.

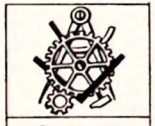
9

Student's t–Test

Using the Program

Procedure	Press	Display/Comments
Clear the memories.	CLR 2nd CMs	0.
Fix the decimal point to desired accuracy (n places).	2nd Fix n	
Enter data pairs.		
First value	−	First value
Second value	A	Pair number
Repeat for each data pair.		
Remove faulty data pair.		
First value (as input).	−	First value
Second value (as input).	2nd A'	Previous pair number
Calculate P.	D	P
Optional:		
Calculate t.	B	$\frac{t}{\Delta x}$
Calculate mean difference.	2nd $\bar{x}$	
Calculate standard deviation of differences.	INV 2nd $\bar{x}$	s

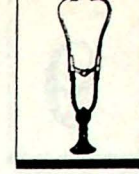
Note: If the information stored in data registers 01-06 by pressing A is already known, it can be entered directly into those registers as shown in *Personal Programming*, p. V-33.



Sample Problem

Suppose the following data were collected from two different systems. Use the program to calculate $\overline{\Delta x}$, s, and P. Is there a significant difference between the two systems?

System 1	*System 2*
6.33	6.14
6.67	6.85
5.42	5.50
7.96	7.43
9.66	8.49
4.42	5.12



Procedure	Press	Display/Comments
Enter program.		0
Clear data registers.	CLR 2nd CMs	0.
Select 3-decimal place display.	2nd fix 3	0.000
Enter data pairs.	6.33 −	6.330
	6.14 A	1.000
	6.67 −	6.670
	6.85 A	2.000
	5.42 −	5.420
	5.5 A	3.000
	7.96 −	7.960
	7.43 A	4.000
	9.66 −	9.660
	8.49 A	5.000
	4.42 −	4.420
	5.12 A	6.000
Calculate mean difference.	2nd $\bar{x}$	0.155 $\overline{\Delta x}$
Calculate standard deviation of difference.	INV 2nd $\bar{x}$	0.644 s
Calculate P.	D	0.581 P

The average difference between the two methods is insignificant.

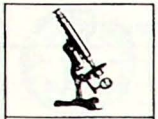
REFERENCES/ INFORMATION

U.S. Department of Commerce, National Bureau of Standards, *Handbook of Mathematical Functions*, M. Abramowitz and Irene Stegun, eds. (Washington, D.C.: U.S. Government Printing Office, 1968).

R. E. Walpole and R. H. Myers, *Probability and Statistics for Engineers* (New York: Macmillan Publishing Co., Inc., 1978).

Personal Programming, pp. V-32ff.

9 Regression Analysis



OBJECTIVE

A second approach for comparing two systems of cardiac output determination is to plot the results as data pairs on a graph. Each cardiac output value from the reference system can be the x-variable of the point and the value from the new system can be the y-variable. A regression line fitted to the data by the calculator will then show a minimum squared-error fit to the values and can be represented by $y = mx + b$ where b is the y-intercept and m is the slope of the line. Additionally, since both systems are expected to yield progressively smaller values as cardiac output approaches zero, a line which is constrained to go through the origin can also be fitted to the data.

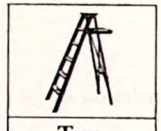
FACTS/ THEORY

The theory applicable to the regression line is described in *Personal Programming*, pp. V-32-39.

One expects that a small value of cardiac output by one method would show a comparably small value by the other. Consequently, a second line is fitted to the data – a line passing through the origin. The equation for this line is $y = m_o x$, where m_o is defined as:

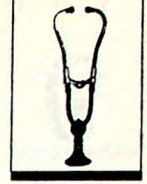
$$m_o = \frac{\Sigma xy}{\Sigma x^2}$$

This is an easy application of the calculator's $\Sigma +$ function because statistical analysis places Σxy in data register 06 and Σx^2 in data register 05. The program in this case simply divides the contents of register 06 by those in register 05.



THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [E] [RCL] 6 [÷] [RCL] 5 [=] [INV] [SBR]	Calculates m_o.
	[LRN] [RST]	Exits learn mode.



USING THE PROGRAM

Procedure	Press	Display/Comments
Clear data registers 01-06.	2nd Pgm 01 SBR CLR	Prepare for statistical storage.
Select 3-decimal display	2nd fix 3	
Enter measurement for method #1, trial 1.	x⇄t	Previous value displayed.
Enter measurement for method #2, trial 1.	2nd Σ+	Pair number displayed.
Repeat this entry sequence for each trial.		
Calculate y mean.	2nd x̄	$\bar{y}$
Calculate x mean.	x⇄t	$\bar{x}$
Calculate standard deviation of y.	INV 2nd x̄	σ_y
Calculate standard deviation of x.	x⇄t	σ_x
Calculate y-intercept.	2nd Op 12	b
Calculate slope.	x⇄t	m
Calculate correlation coefficient.	2nd Op 13	R*
Calculate slope m_o for the equation $y = m_o x$.	E	m_o*

*This value is printed if the PC-100A is used.

9

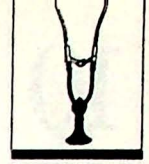
Regression Analysis

Sample Problem

Apply the regression analysis to the data in the previous section, listed again below.

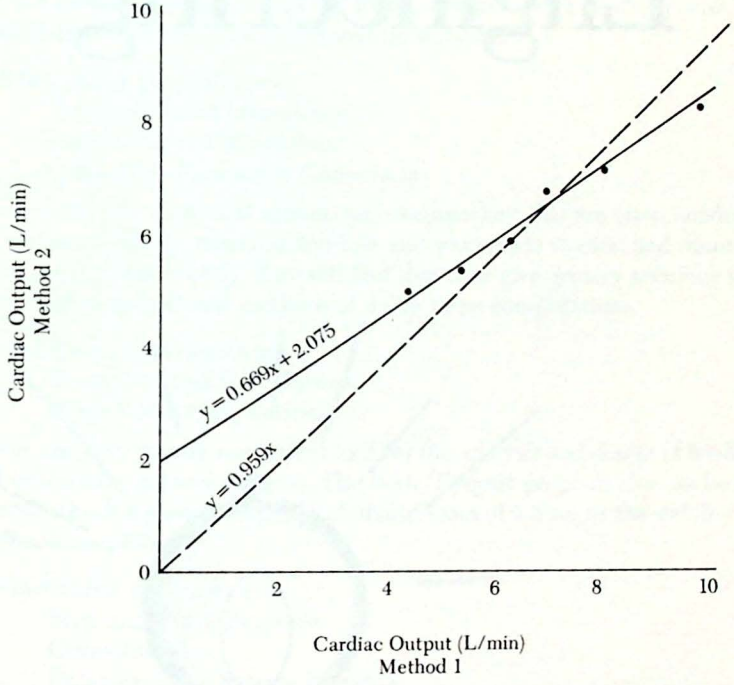
Trial	*System #1*	*System #2*
1	6.33	6.14
2	6.67	6.85
3	5.42	5.50
4	7.96	7.43
5	9.66	8.49
6	4.42	5.12

Procedure	Press	Display/Comments
Clear data memories 01-06.	2nd Pgm 01 SBR CLR	0.
Select 3-decimal display.	2nd Fix 3	0.000
Enter data points.	6.33 x:t	0.000
	6.14 2nd Σ+	1.000
	6.67 x:t	7.330
	6.85 2nd Σ+	2.000
	5.42 x:t	7.670
	5.5 2nd Σ+	3.000
	7.96 x:t	6.420
	7.43 2nd Σ+	4.000
	9.66 x:t	8.960
	8.49 2nd Σ+	5.000
	4.42 x:t	10.660
	5.12 2nd Σ+	6.000
Calculate statistics.	2nd $\bar{x}$	6.588 $\bar{y}$
	x:t	6.743 $\bar{x}$
	INV 2nd $\bar{x}$	1.260 σ_y
	x:t	1.861 σ_x
	2nd Op 12	2.075 b
	x:t	0.669 m
	2nd Op 13	0.989 R
Calculate slope m_o.	E	0.959 m_o



The regression line computed by the calculator is defined as $y = mx + b = 0.669x + 2.075$. The mathematical representation of the line constrained to the origin and the data points is $y = m_o x = 0.959x$.

A plot of the input data and the resulting regression lines is shown below.



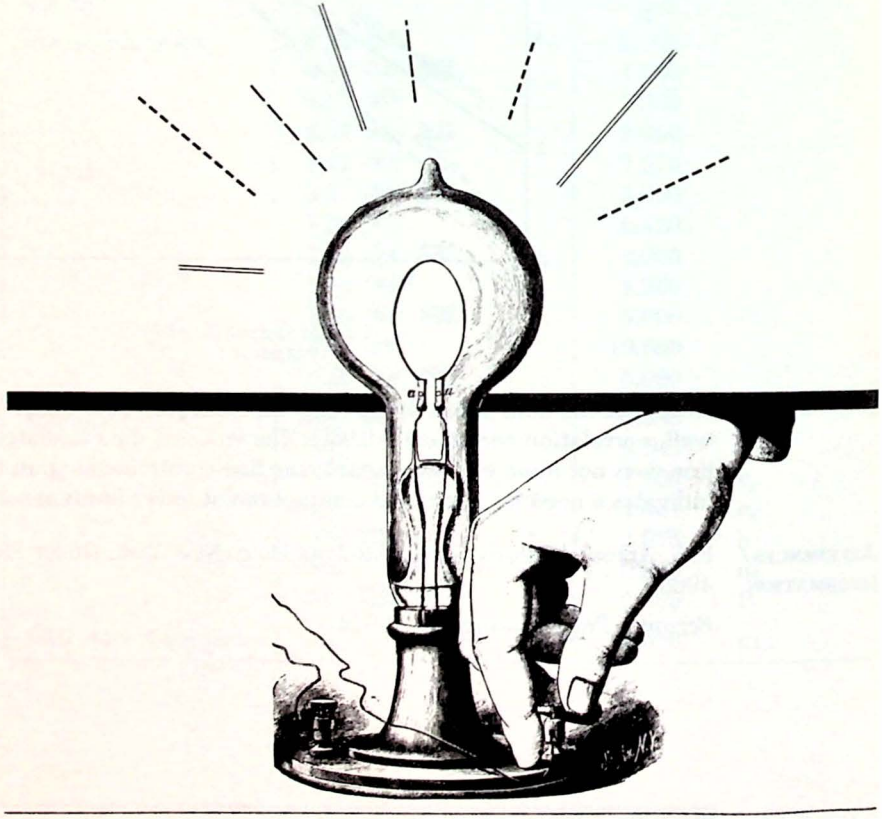
Note that the data points fit the regression line plotted through them quite well (correlation coefficient = 0.959). The fact that the calculated regression line does not more closely resemble the line constrained to pass through zero indicates a need for additional comparisons at lower levels of cardiac output.

REFERENCES/ INFORMATION

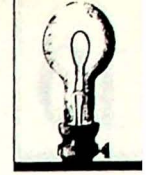
F. S. Acton, *Analysis of Straight-Line Data* (New York: Dover Publishing Co., 1966).

Personal Programming, pp. V-32ff.

10 Electrical and Systems Engineering



Introduction



This chapter contains eight programs of interest in electrical engineering and the related field of systems engineering. The *Facts/Theory* section for each program summarizes the necessary background information but assumes that you have some familiarity with the subject. In other words, these programs will help you learn more about electrical and systems engineering, but they are not intended to teach you the subjects from scratch. You may wish to consult the references for additional information.

The first three programs are:

Series-Parallel Impedance
Smith Chart Calculations
Transistor Parameter Conversion

These carry out practical numerical computations that are often needed in AC circuit analysis, transmission-line and waveguide studies, and transistor electronics, respectively. You will find that they give greater accuracy in less time than conventional methods of doing these computations.

The next two programs are:

Bode/Nyquist Calculations
Root-Locus Calculations

These are very handy numerical aids for the analysis and design of feedback and automatic control systems. The Bode/Nyquist program also can be used to study the frequency-response characteristics of a filter or the stability of a feedback amplifier.

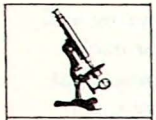
The last three programs are:

Step and Pulse Response
Convolution
Discrete-Time System Response

All of these are simulation programs in the sense that they numerically simulate the time response of a linear system – electrical, mechanical, or whatever. Over and above their value for analysis work, these programs will help you get a better feeling for the relationship between the mathematical model of a system and its actual physical behavior.

As a student or practitioner of electrical engineering, you will probably find at least some of these programs to be relevant to your day-to-day work. No doubt you will also want to write other programs to tackle problems and computations not covered here. In that regard, the programs given here may suggest some algorithms and programming techniques that will be of help to you.

10 Series–Parallel Impedance



OBJECTIVE

In determining the input or output impedance of an AC circuit, it is often necessary to reduce a group of resistors, capacitors, and inductors into a single equivalent complex impedance. This program will enable you to analyze highly complicated passive circuits in a quick and easy manner if all elements are connected in series and parallel.

FACTS/ THEORY

Passive networks consisting only of impedances can be reduced to a single equivalent impedance. To combine two impedances in series, you add them to get the total equivalent impedance.

$$Z = Z_A + Z_B$$

If two impedances are connected in parallel, the total equivalent impedance is

$$Z = Z_A \| Z_B = \left[\frac{1}{Z_A} + \frac{1}{Z_B}\right]^{-1} = \frac{Z_A Z_B}{Z_A + Z_B}$$

By repeatedly using these two methods, you can reduce a group of impedances to one equivalent impedance.

In general, an impedance is a complex quantity of the form $Z = R + jX$. This program uses Master Library program ML-04 to perform the complex arithmetic.

Prior to using the program, all impedances in series in any branch should be combined into rectangular form $(R + jX)$ by separately summing the real and imaginary parts. In using the program, you should start by entering the impedance most distant from the input terminals.

THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 9 R/S STO 10 R/S	Stores impedance $Z_A = (R_A + jX_A)$. R_A stored in memory 9. X_A stored in memory 10.
008	2nd Lbl B STO 5 R/S STO 6 R/S	Stores impedance $Z_B = (R_B + jX_B)$. R_B stored in memory 5. X_B stored in memory 6.

Step #	Program Keystrokes	Purpose/Comments
016	2nd Lbl 2nd B′ STO 7 R/S STO 8 R/S	Stores impedance $Z_{B'} = (R_{B'} + jX_{B'})$. $R_{B'}$ stored in memory 7. $X_{B'}$ stored in memory 8.
024	SBR 2nd Prd RCL 1 STO 5 R/S RCL 2 STO 6 R/S	Calls subroutine Prd to compute $Z_B \| Z_{B'}$ and stores the result in memories 5 and 6.
036	2nd Lbl C RCL 9 STO 7 RCL 10 STO 8 SBR 2nd Prd RCL 1 STO 9 R/S RCL 2 STO 10 R/S	Stores Z_A in memories 7 and 8. Calls subroutine Prd to compute $Z_A \| Z_B$ and stores result in memories 9 and 10.
058	2nd Lbl 2nd Prd RCL 5 STO 1 RCL 6 STO 2 RCL 7 STO 3 RCL 8 STO 4 2nd Pgm 04 C	Recalls impedances to be used and places them in R_{01}, R_{02}, R_{03}, and R_{04} and calls program ML-04 to compute $Z_B \times Z_{B'}$.
079	RCL 5 + RCL 7 = STO 3 RCL 6 + RCL 8 = STO 4 2nd Pgm 04 2nd C′ INV SBR	Computes $Z_B + Z_{B'}$ and stores result in memories 3 and 4. Calls program ML-04 to complete computation of $Z_B \| Z_{B'}$.
099	2nd Lbl D RCL 5 SUM 9 RCL 9 R/S RCL 6 SUM 10 RCL 10 R/S	Computes $Z_A + Z_B$ and stores results in memory 5 and 6.
115	2nd Lbl E RCL 9 STO 1 RCL 10 STO 2 2nd Pgm 05 B STO 11 $x \rightleftarrows t$ × 180 ÷ 2nd π = STO 12 RCL 11 R/S RCL 12 R/S	Recalls Z_A and stores it in memories 1 and 2. Calls program ML-05 to compute $\|Z_A\|$ and angle of Z_A in degrees. Recalls and displays results.
	LRN RST	Exits learn mode.

10

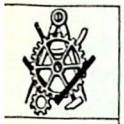
Series—Parallel Impedance

Register Contents

R_{00}		R_{09}	R_A
$R_{01} - R_{04}$	ML-04, ML-05	R_{10}	X_A
R_{05}	R_B	R_{11}	$\|Z_A\|$
R_{06}	X_B	R_{12}	$\angle Z_A$
R_{07}	$R_{B'}$		
R_{08}	$X_{B'}$		

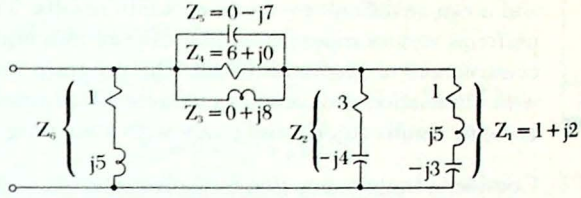
Using the Program

Procedure	Press	Display/Comments
Enter most "distant" impedance, $Z_A = (R_A + jX_A)$.		
R_A	A	R_A
X_A	R/S	X_A
Enter next impedance to be combined, Z_B.		
R_B	B	R_B
X_B	R/S	X_B
If the next impedance is in parallel with Z_B, enter $Z_{B'}$.		
$R_{B'}$	2nd B'	$R_{B'}$
$X_{B'}$	R/S	$X_{B'}$
Compute new $Z_B = Z_B \parallel Z_{B'}$.	R/S	New R_B.
	R/S	New X_B.
Repeat until all impedances have been combined in parallel with Z_B.		
Compute new $Z_A = Z_A \parallel Z_B$.	C	New R_A.
	R/S	New X_A.
OR		
Compute new $Z_A = Z_A + Z_B$.	D	New R_A.
	R/S	New X_A.
Compute $\|Z_A\|$ and the angle of Z_A.	E	$\|Z_A\|$
	R/S	$\angle Z_A$



SAMPLE PROBLEM

Find the total equivalent impedance for the following circuit.



Procedure	Press	Display/Comments
Enter program.		0
Fix decimal for desired accuracy.	2nd fix 2	0.00
Enter most distant impedance, Z_1.	1 A 2 R/S	1.00 R_1 2.00 X_1
Enter next impedance, Z_2.	3 B 4 +/− R/S	3.00 R_2 −4.00 X_2
Compute new $Z_A = Z_1 \| Z_2$.	C R/S	2.00 R_A 1.50 X_A
Enter next impedance, Z_3. (Choose any of the three.)	0 B 8 R/S	0.00 R_3 8.00 X_3
Enter next impedance, Z_4.	6 2nd B' 0 R/S	6.00 R_4 0.00 X_4
Compute new $Z_B = Z_3 \| Z_4$.	R/S R/S	3.84 R_B 2.88 X_B
Enter next impedance, Z_5.	0 2nd B' 7 +/− R/S	0.00 R_5 −7.00 X_5
Compute new $Z_B = Z_B \| Z_5$.	R/S R/S	5.93 R_B −0.64 X_B
Compute new $Z_A = Z_A + Z_B$.	D R/S	7.93 R_A 0.86 X_A
Enter last impedance, Z_6.	10 B 5 R/S	10.00 R_6 5.00 X_6
Compute new $Z_A = Z_6 \| Z_B$.	C R/S	4.57 R_A 1.20 X_A
Display polar form if desired.	E R/S	4.73 $\|Z_A\|$ 14.67 $\angle Z_A$

10 Smith Chart Calculations

OBJECTIVE

Graphical constructions on the Smith Chart require some special techniques, and it can be difficult to obtain accurate results. The program given here performs various transmission-line calculations equivalent to the graphical constructions on the Smith Chart. The program makes provisions for lines with attenuation and complex characteristic impedance. Thus you can obtain accurate results quickly and easily with your programmable calculator.

FACTS/ THEORY

Consider a transmission line with characteristic impedance Z_o (possibly complex). The normalized impedance (z) at any point will be the impedance Z at that point divided by Z_o ($z = Z/Z_o$). Let the normalized distance in wavelengths from the load end of the line be $x = d/\lambda$. Then the per-wavelength phase constant is given by

$$b = \beta\lambda = 2\pi \text{ (radians)}$$

and the attenuation constant is

$$a = 8.686\alpha\lambda \text{ (dB per wavelength)}$$

where α is the attenuation factor (nepers per unit length).

This program has numerous subroutines which will perform different calculations depending on the various inputs. The following terms are used in many equations and will be defined here for convenience.

Z_o	—	characteristic impedance
Z	—	impedance°
z	—	normalized impedance°
Y	—	admittance°
P	—	reflection factor° (rectangular coordinates)
$\lvert P\rvert, \measuredangle P$	—	reflection factor° (polar coordinates)
x_{min}	—	position of the first voltage minimum
σ	—	voltage standing wave ratio (VSWR)

°Note that a subscript of "L" indicates values at termination, and a subscript of "i" indicates values at input.

Slotted-Line Measurement of Termination Impedance

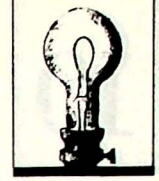
Assuming a low-loss line, the termination impedance

$$Z_L = \text{Re}(Z_L) + j\ \text{Im}(Z_L)$$

is calculated by the following sequence.

$$|P_L| = \frac{\sigma - 1}{\sigma + 1} \qquad \measuredangle P_L = \pi + 4\pi x_{min}$$

$$z_L = \frac{1 + P_L}{1 - P_L} \qquad Z_L = Z_o z_L$$



REFLECTION FACTOR AT TERMINATION

Given the termination impedance

$$Z_L = \mathrm{Re}(Z_L) + j\ \mathrm{Im}(Z_L)$$

or admittance

$$Y_L = \mathrm{Re}(Y_L) + j\ \mathrm{Im}(Y_L)$$

the complex reflection factor is calculated as follows.

$$Z_L = \frac{1}{Y_L} \qquad \text{(if } Y_L \text{ entered)}$$

$$z_L = \frac{Z_L}{Z_o} \qquad P_L = \frac{z_L - 1}{z_L + 1}$$

INPUT IMPEDANCE

Given Z_L and P_L from a previous calculation, the input impedance

$$Z_i = \mathrm{Re}(Z_i) + j\ \mathrm{Im}(Z_i)$$

at distance x from the termination is computed by the following sequence.

$$|P_i| = |P_L|\ (10^{-0.1a})^x \qquad \measuredangle P_i = \measuredangle P_L - 4\pi x$$

$$z_i = \frac{1 + P_i}{1 - P_i} \qquad Z_i = Z_o z_i$$

Note that this sequence applies in the reverse direction (from Z_i to Z_L) if x is entered as a negative quantity.

ADMITTANCE, VSWR, AND VOLTAGE MINIMUM

The following calculations are the same for either input or termination values.

$$Y = \frac{1}{Z} \qquad \sigma = \frac{1 + |P|}{1 - |P|}$$

$$x_{min} = \frac{1}{4\pi}(\measuredangle P - \pi)$$

EQUIVALENT SHORT-CIRCUITED LINE

Given Z_L from a previous entry or calculation, the length x_{scl} of a low-loss short-circuited line whose input impedance equals Z_L is given by

$$x_{scl} = \frac{1}{2\pi}\tan^{-1}[\mathrm{Im}(z_L) - j\mathrm{Re}(z_L)]$$

Note that, if $\mathrm{Im}(x_{scl}) \neq 0$, there is no exact equivalent.

SINGLE-STUB IMPEDANCE MATCH

Given P_L from a previous calculation, the following equations compute the insertion position, x_{line}, and the stub length, x_{stub}, such that $Z = Z_o$ at $x = x_{line}$.

$$x_{line} = \frac{1}{4\pi}\left[P_L + \cos^{-1}(-|P_L|)\right]$$

$$|P'| = |P_L|(10^{-0.1a})^{x_{line}}$$

$$\measuredangle P' = \measuredangle P_L - 4\pi x_{line}$$

$$z' = \frac{1+P'}{1-P'}$$

$$x_{stub} = \frac{1}{2\pi}\tan^{-1}\left\{\left[\mathrm{Im}(1/z')\right]^{-1}\right\}$$

Note that the stub is assumed to have the same parameters as the line, which is assumed to be low-loss.

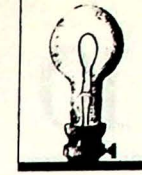
The following program is designed to do these calculations for you. The full program exceeds the capacity of the *TI Programmable 58* calculator. However, by omitting subroutine [E'], which computes x_{line}, x_{stub}, the program can be used on the *TI Programmable 58* calculator if you repartition the calculator (press 2 [2nd] [Op] 17) before you key in the program.

THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enter learn mode.
000	[2nd] [Lbl] [2nd] [A'] [2nd] [CMs] [2nd] [Rad] 4 [×] [2nd] [π] [=] [STO] 19 1 [STO] 12 [STO] 13 [R/S]	Initialize routine. Clears memories and puts calculator in radian mode. Stores 4π in R_{19}. Enters default values $10^{-0.1a} = 1$, $\mathrm{Re}(Z_o) = 1$.
016	[STO] 13 [R/S] [STO] 14 [R/S] [×] .1 [=] [+/−] [INV] [2nd] [log] [STO] 12 [R/S]	Stores Re (Z_o) in R_{13} and $\mathrm{Im}(Z_o)$ in R_{14}. Computes $10^{-0.1a}$ and stores it in R_{12}.
032	[2nd] [Lbl] [A] [STO] 9 [−] 1 [=] [STO] 15 [RCL] 9 [+] 1 [=] [INV] [2nd] [Prd] 15 [R/S]	Computes Z. Stores σ in R_9. Computes $\lvert P_L\rvert = (\sigma-1)/(\sigma+1)$ and stores it in R_{15}.

Step #	Program Keystrokes	Purpose/Comments
050	STO 10 × RCL 19 + 2nd π = STO 16	Computes $\measuredangle P_L = \pi + 4\pi x_{min}$ and stores it in R_{16}.
060	2nd Lbl 2nd P→R RCL 15 x:t RCL 16 2nd P→R STO 6 STO 2 +/− STO 4 x:t STO 5 STO 1 +/− STO 3 2nd Op 21 2nd Op 23	Converts P to rectangular coordinates. Stores Im (P) in R_2 and R_6. Stores −Im (P) in R_4. Stores Re (P) in R_5. Stores 1+Re(P) in R_1 and 1−Re(P) in R_3.
087	2nd Pgm 04 2nd C' STO 7 x:t STO 08 RCL 13 STO 3 RCL 14 STO 4 2nd Pgm 04 C STO 17 INV SBR x:t STO 18 R/S	Computes z=(1+P)/(1−P). Stores z in R_7, R_8. Recalls Z_o and computes $Z = zZ_o$. Stores Z in R_{17}, R_{18}.
113	2nd Lbl B STO 17 STO 1 R/S STO 18 STO 02	Computes P_L. Stores Z_L.
124	2nd Ind ÷ RCL 13 STO 3 RCL 14 STO 4 2nd Pgm 04 2nd C' STO 7 STO 3 x:t STO 8 STO 4 2nd Op 31 2nd Op 23	Recalls Z_o and stores it for complex division. Calls program ML-04 to compute $z_L = Z_L/Z_o$. Stores z_L. Stores Re $(z_L)-1$ in R_1 and Re $(z_L)+1$ in R_3.
150	2nd Pgm 04 2nd C' STO 5 x:t STO 6 INV 2nd P→R STO 16 x:t STO 15 GTO E	Computes $P_L = (z_L - 1) \div (z_L + 1)$ and stores it in R_5, R_6. Converts P_L to polar coordinates and stores $\measuredangle P_L$ in R_{16} and $\lvert P_L \rvert$ in R_{15}.
167	2nd Lbl 2nd B' STO 1 R/S STO 2 2nd Pgm 05 E STO 17 x:t STO 18 GTO ÷	Computes Z=1/Y. Stores Y in R_1, R_2. Calls ML-05 to compute Z=1/Y. Stores Z in R_{17}, R_{18}. Transfers to ÷ to compute P.
184	2nd Lbl 2nd C' RCL 17 STO 1 RCL 18 STO 2 2nd Pgm 05 E R/S x:t R/S	Computes Y=1/Z. Stores Z in R_1, R_2. Calls ML-05 to compute Y=1/Z. Displays Y.

Step #	Program Keystrokes	Purpose/Comments
200	2nd Lbl C STO 10 × RCL 19 = INV SUM 16 (RCL 12 y^x RCL 10) 2nd Prd 15 GTO 2nd P→R	Computes input impedance at x. Stores x. Computes $\measuredangle P_i = \measuredangle P_L - 4\pi x$. Computes $\lvert P_i\rvert = \lvert P_L\rvert (10^{-0.1a})^x$. Transfers to subroutine P→R to compute Z_i.
222	2nd Lbl D RCL 15 + 1 = STO 9 +/− + 2 = INV 2nd Prd 9 RCL 9 R/S	Computes σ and x_{min}. Computes $\sigma = \frac{1+\lvert P\rvert}{1-\lvert P\rvert}$ and stores it in R_9.
241	RCL 16 − 2nd π = ÷ RCL 19 = SBR 2nd x≥t STO 10 R/S	Computes $x_{min} = (\measuredangle P - \pi) / 4\pi$. Calls subroutine x≥t to force $x_{min} \geq 0$. Stores x_{min} in R_{10}.
255	2nd Lbl 2nd x≥t 2nd CP 2nd x≥t 2nd |x| + .5 = GTO 2nd x≥t 2nd Lbl 2nd |x| INV SBR	If $x \geq 0$, transfers to 2nd |x|. If $x < 0$, adds 0.5 and repeats testing until $x \geq 0$.
269	2nd Lbl E RCL 15 R/S RCL 16 × 180 ÷ 2nd π = R/S	Displays P. Displays |P|. Converts $\measuredangle P$ from radians to degrees and displays $\measuredangle P$.
284	2nd Lbl 2nd D' RCL 8 STO 1 RCL 7 +/− STO 2 2nd Pgm 06 2nd D' ÷ 2 ÷ 2nd π = SBR 2nd x≥t STO 10 R/S	Computes x_{sc1}. Stores Im (z_L) in R_1 and $-$Re (z_L) in R_2. Calls ML-06 to compute $\tan^{-1}[\mathrm{Im}(z_L) - j\,\mathrm{Re}(z_L)]$. Divides Re (x_{sc1}) by 2π. Calls subroutine x≥t to force Re $(x_{sc1}) \geq 0$. Stores Re (x_{sc1}) in R_{10}.
308	RCL 2 ÷ 2 ÷ 2nd π = SBR 2nd x≥t STO 11 R/S	Divides Im (x_{sc1}) by 2π. Calls subroutine x≥t to force Im $(x_{sc1}) \geq 0$. Stores Im (x_{sc1}) in R_{11}.

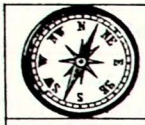


Step #	Program Keystrokes	Purpose/Comments
320	2nd Lbl 2nd E RCL 15 +/− INV 2nd cos + RCL 16 = ÷ RCL 19 = SBR 2nd x≥t STO 10 R/S	Computes x_{line} and x_{stub}. Computes $x_{line} = \frac{\cos^{-1}(-\lvert P_L \rvert)}{4\pi}$ Calls subroutine x≥t to force $x_{line} \geq 0$. Stores x_{line} in R_{10}.
340	C RCL 7 STO 1 RCL 8 STO 2 2nd Pgm 05 E x:t 1/x INV 2nd tan ÷ 2 ÷ 2nd π = SBR 2nd x≥t STO 11 R/S	Calls subroutine C to compute z′. Stores z′ in R_1, R_2. Calls ML-05 to compute 1/z′. Computes $x_{stub} = \frac{\tan^{-1}[\mathrm{Im}(1/z')]^{-1}}{2\pi}$ Calls subroutine x≥t to force $x_{stub} \geq 0$. Stores x_{stub} in R_{11}.
	LRN RST	Exits learn mode.

REGISTER CONTENTS

R_{00}		R_{10}	x, x_{min}, x_{scl}, x_{line}
R_{01}	Used	R_{11}	Im (x_{scl}), x_{stub}
R_{02}	Used	R_{12}	$10^{-0.1a}$
R_{03}	Used	R_{13}	Re (Z_o)
R_{04}	Used	R_{14}	Im (Z_o)
R_{05}	Re (P)	R_{15}	|P|
R_{06}	Im (P)	R_{16}	$\angle$ P
R_{07}	Re (z)	R_{17}	Re (Z)
R_{08}	Im (z)	R_{18}	Im (Z)
R_{09}	σ	R_{19}	4π

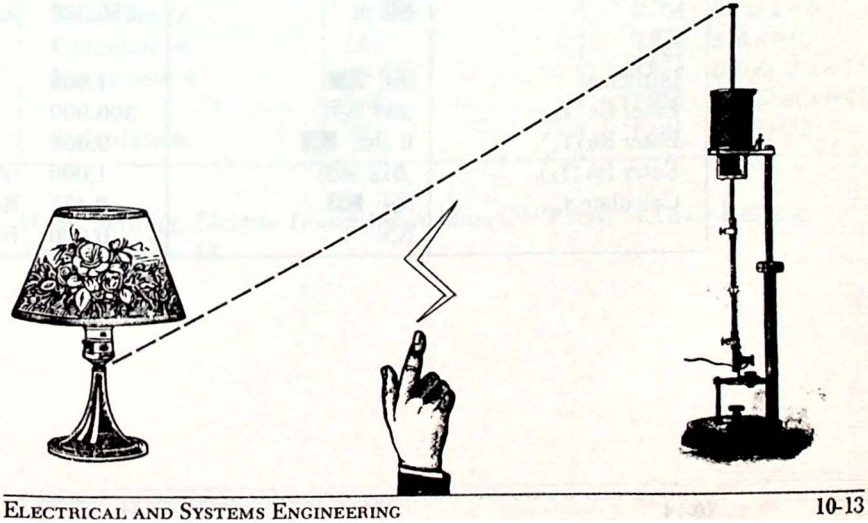
10 Smith Chart Calculations



Using the Program

Procedure	Press	Display/Comments
Initialize.	2nd A'	1.
Enter Re (Z_o).	Re (Z_o) R/S	Re (Z_o)
Enter Im (Z_o).	Im (Z_o) R/S	Im (Z_o)
Enter attenuation, a, in dB.	a R/S	$10^{-0.1a}$
Note: the above three steps may be omitted if you are using the default values for Z_o and a.		
To calculate Z_L from σ and x_{min}.		
Enter σ.	σ A	$(\sigma+1)$
Enter x_{min}.	x_{min} R/S	Re (Z_L)
	R/S	Im (Z_L)
To calculate P from Z.		
Enter Re (Z).	Re (Z) B	Re (Z)
Enter Im (Z).	Im (Z) R/S	|P|
	R/S	$\angle$P in degrees.
To compute P from Y.		
Enter Re (Y).	Re (Y) 2nd B'	Re (Y)
Enter Im (Y).	Im (Y) R/S	|P|
	R/S	$\angle$P in degrees.
To calculate Y. Note that Z_L should be stored in R_{17}, R_{18} before using this subroutine. May be used after A or B.	2nd C'	Re (Y)
	R/S	Im (Y)
To find Z_i from x. Note that this subroutine should be used after A, B, or B'.		
Enter x.	x C	Re (Z_i)
	R/S	Im (Z_i)

Procedure	Press	Display/Comments
To compute σ and x_{min}. Note that this subroutine should be used after [B], [B'], or [C].	[D]	σ
	[R/S]	x_{min}
To calculate P. Note that this subroutine should be used after [A] or [C].	[E]	$\|P\|$
	[R/S]	$\angle P$ in degrees.
To calculate x_{scl}. Note this subroutine should be used after [A], [B], [B'], or [C].	[2nd] [D']	Re (x_{scl})
	[R/S]	Im (x_{scl})
To calculate x_{line} and x_{stub}. Note that this subroutine should be used after [A], [B], or [B'].	[2nd] [E']	x_{line}
	[R/S]	x_{stub}



10 Smith Chart Calculations

Sample Problem

(a) Find Z_L and P_L given $Z_o = 50 + j0$, $a = 0$, $\sigma = 3.7$, and $x_{min} = 0.106$.
(b) Find x_{scl} given $Z_o = 300 + j0$, $a = 0$, and $Y_L = 0 + j0.012$.
(c) Find σ and x_{min}, and Y_i at $x = 0.2$, given $Z_o = 50 + j0$, $a = 0$, and $Z_L = 30 + j20$.
(d) Find x_{line} and x_{stub} given $Z_o = 50 + j0$, $a = 0$, and $Y_L = 0.015 - j0.025$.
(e) Find σ at $x = 0$ and Z and σ at $x = 15$ given $Z_o = 50 + j0$, $a = 0.4$, and $Z_L = 10 + j40$.

Procedure	Press	Display/Comments
Enter program.		0
Fix decimal for desired accuracy (we'll use 3).	2nd fix 3	0.000
(a)		
Initialize.	2nd A'	1.000
Enter Re (Z_o).	50 R/S	50.000
Since Im (Z_o) and a both equal zero (default values), we need not enter these.		
Enter σ.	3.7 A	4.700 $(\sigma + 1)$
Enter x_{min}.	.106 R/S	20.914 Re (Z)
	R/S	−34.848 Im (Z)
Compute P_L.	E	0.574 $\lvert P_L \rvert$
	R/S	256.320 $\measuredangle P_L$ in degrees.
(b)		
Initialize.	2nd A'	1.000
Enter Re (Z_o).	300 R/S	300.000
Enter Re (Y_L).	0 2nd B'	0.000
Enter Im (Y_L).	.012 R/S	1.000 $\lvert P_L \rvert$
Calculate x_{scl}.	2nd D'	0.457 Re (x_{scl})
	R/S	0.000 Im (x_{scl})

Procedure	Press	Display/Comments
(c)		
Initialize.	2nd A'	1.000
Enter Re (Z_o).	50 R/S	50.00
Enter Re (Z_L).	30 B	30.000
Enter Im (Z_L).	20 R/S	0.343 $\lvert P_L \rvert$
Compute σ.	D	2.044 σ
Compute x_{min}.	R/S	0.418 x_{min}
Enter x_i and	.2 C	90.711 Re (Z_i)
compute Z_i.	R/S	−27.597 Im (Z_i)
Compute Y_i.	2nd C'	0.010 Re (Y_i)
	R/S	0.003 Im (Y_i)
(d)		
Initialize.	2nd A'	1.000
Enter Re (Z_o).	50 R/S	50.000
Enter Re (Y_L).	.015 2nd B'	0.015
Enter Im (Y_L).	.025 +/− R/S	0.593 |P|
Compute x_{line}.	2nd E'	0.334 x_{line}
Compute x_{stub}.	R/S	0.095 x_{stub}
(e)		
Initialize.	2nd A'	1.000
Enter Re (Z_o).	50 R/S	50.000
Enter Im (Z_o).	0 R/S	0.000
Enter a.	.4 R/S	0.912 $10^{-.04}$
Enter Re (Z_L).	10 B	10.000
Enter Im (Z_L).	40 R/S	0.784 |P| at x = 0.
Calculate σ.	D	8.279 σ at x = 0.
Enter new x.	15 C	43.059 Re (Z) at x = 15.
	R/S	17.312 Im (Z) at x = 15.
Calculate σ.	D	1.491 σ at x = 15.

REFERENCES/ INFORMATION

H. H. Skilling, *Electric Transmission Lines* (New York: McGraw-Hill, Inc., 1951), chapter 12.

10 Transistor Parameter Conversion

OBJECTIVE

Since manufacturers' data sheets often provide the low-frequency (real) hybrid parameters of a transistor in only one configuration (CE, CB, or CC), it may be necessary to convert the h-parameters from one configuration to another. The following program will perform these conversions for you.

FACTS/ THEORY

The hybrid or h-parameters are defined as

$$v_1 = h_{11}i_1 + h_{12}v_2$$
$$i_2 = h_{21}i_1 + h_{22}v_2$$

For a common base (CB) configuration, the equations are

$$v_{eb} = h_{ib}i_e + h_{rb}v_{cb}$$
$$i_c = h_{fb}i_e + h_{ob}v_{cb}$$

The equations for a common emitter (CE) configuration are

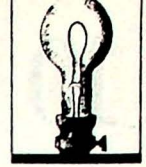
$$v_{be} = h_{ie}i_b + h_{re}v_{ce}$$
$$i_c = h_{fe}i_b + h_{oe}v_{ce}$$

For a common collector (CC) configuration, the equations are

$$v_{bc} = h_{ic}i_b + h_{oc}v_{ec}$$
$$i_e = h_{fc}i_b + h_{oc}v_{ec}$$

The program listed below utilizes four subroutines to make the necessary conversions:

Subroutine	Conversions	Calculations Performed
A	Common Collector to Common Emitter	$h_{ie} = h_{ic}$ $h_{re} = 1 - h_{rc}$ $h_{fe} = -(1 + h_{fc})$ $h_{oe} = h_{oc}$
B	Common Emitter to Common Base	$h_{ib} = \frac{h_{ie}}{D}$ $h_{rb} = \frac{h_{ie}h_{oe} - h_{re}(1 + h_{fe})}{D}$ $h_{fb} = \frac{-h_{fe}(1 - h_{re}) - h_{ie}h_{oe}}{D}$ $h_{ob} = \frac{h_{oe}}{D}$ where $D = (1 + h_{fe})(1 - h_{re}) + h_{ie}h_{oe}$



C	Common Base to Common Collector	$h_{ic}=\frac{h_{ib}}{D}$ $h_{rc}=\frac{1+h_{fb}}{D}$ $h_{fc}=\frac{h_{rb}-1}{D}$ $h_{oc}=\frac{h_{ob}}{D}$ where $D=(1+h_{fb})(1-h_{rb})+h_{ob}h_{ib}$
D	Common Collector to Common Base	$h_{ib}=\frac{h_{ic}}{D}$ $h_{rb}=\frac{h_{fc}(1-h_{rc})+h_{ic}h_{oc}}{D}$ $h_{fb}=\frac{h_{rc}(1+h_{fc})-h_{ic}h_{oc}}{D}$ $h_{ob}=\frac{h_{oc}}{D}$ where $D=h_{ic}h_{oc}-h_{fc}h_{rc}$

To use the program you must enter the h-parameters you know in the order h_{11}, h_{12}, h_{21}, and h_{22}. Results will be displayed in the same order.

THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl 2nd B' RCL 5 INV 2nd Prd 1 INV 2nd Prd 2 INV 2nd Prd 3 INV 2nd Prd 04	Divides by D to find h_{ib}, h_{rb}, h_{fb}, h_{ob}, and h_{ic}, h_{rc}, h_{fc}, h_{oc}.
016	2nd Lbl 2nd A' RCL 1 2nd Prt R/S	Displays h_{11} and stops.
022	RCL 2 2nd Prt R/S	Displays h_{12} and stops.
026	RCL 3 2nd Prt R/S	Displays h_{21} and stops.
030	RCL 4 2nd Prt R/S	Displays h_{22} and stops.

Transistor Parameter Conversion

Step #	Program Keystrokes	Purpose/Comments
034	2nd Lbl 2nd C' 1 + RCL 3 = × (1 − RCL 2) + RCL 1 × RCL 4 = STO 5 INV SBR	Calculates divisor (D) for subroutines B and C.
058	2nd Lbl E CLR 2nd CMs EE 2nd Fix 2 R/S	Initializes program by clearing display and memories and selecting scientific notation and fix-2 display formats.
066	STO 1 R/S STO 2 R/S STO 3 R/S STO 4 R/S	Stores beginning data, h_{11}, h_{12}, h_{21}, h_{22}. ***Note:*** *Data must be entered in this order.*
078	2nd Lbl A 1 − RCL 2 = STO 2 RCL 3 + 1 = +/− STO 3 GTO 2nd A'	Converts CC data to CE configuration. Transfers to A' to recall new values from memories 1 through 4.
097	2nd Lbl B 2nd C' RCL 1 × RCL 4 − RCL 2 × (1 + RCL 3 = STO 6 RCL 3 +/− × (1 − RCL 2) − RCL 1 × RCL 4 = STO 3 RCL 6 STO 2 GTO 2nd B'	Converts CE to CB configuration.
142	2nd Lbl C 2nd C' INV 2nd Prd 1 INV 2nd Prd 4 RCL 3 + 1 = ÷ RCL 5 = STO 6 RCL 2 − 1 = ÷ RCL 5 = STO 3 RCL 6 STO 2 GTO 2nd A'	Converts CB to CC.

Step #	Program Keystrokes	Purpose/Comments
179	2nd Lbl D RCL 1 × RCL 4 − RCL 3 × RCL 2 = STO 5 RCL 3 × (1 − RCL 2) + RCL 1 × RCL 4 = STO 6 RCL 2 × (1 + RCL 3) − RCL 1 × RCL 4 = STO 3 RCL 6 STO 2 GTO 2nd B′	Converts CC to CB. Steps 181-194 calculate and store divisor (D) for later use in the division subroutine, B′.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}		R_{05}	D
R_{01}	h_{11}	R_{06}	Used
R_{02}	h_{12}	R_{07}	
R_{03}	h_{21}	R_{08}	
R_{04}	h_{22}	R_{09}	

USING THE PROGRAM

Procedure	Press	Display/Comments
Initialize program.	E	0.00 00 Clears display and data registers. Sets scientific notation and 2-decimal modes.
Enter data.		
h_{11}	R/S	h_{11}
h_{12}	R/S	h_{12}
h_{21}	R/S	h_{21}
h_{22}	R/S	h_{22}
Select desired conversion.	A or B or C or D	
	R/S	h_{11}° Converted value.
	R/S	h_{12}° Converted value.
	R/S	h_{21}° Converted value.
	R/S	h_{22}° Converted value.

°This value is printed if the PC-100A is used.

10

Transistor Parameter Conversion

Sample Problem

Given the CE hybrid parameters

$h_{ie} = 1.7k\Omega$ $\quad h_{fe} = 180$

$h_{re} = 2.5x10^{-4}$ $\quad h_{oe} = 2.3x10^{-6}$ siemens

find the CB and CC parameters. (Refer to *Facts/Theory* section for a discussion of the following equations.)

$$v_{be} = h_{ie}i_b + h_{re}v_{ce}$$
$$i_c = h_{fe}i_b + h_{oe}v_{ce}$$

Using matrix notation, these become

$$\begin{vmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{vmatrix} = \begin{vmatrix} h_{ie} & h_{re} \\ h_{fe} & h_{oe} \end{vmatrix} = \begin{vmatrix} 1700 & 2.5x10^{-4} \\ 180 & 2.3x10^{-6} \end{vmatrix}$$

Procedure	Press	Display/Comments
Enter program.		0
Initialize program.	E	0.00 00
Enter h values in CE configuration.		
h_{11}	1700 R/S	1.70 03
h_{12}	2.5 EE	2.5 00
	4 +/− R/S	2.50 −04
h_{21}	180 R/S	1.80 02
h_{22}	2.3 EE	2.3 00
	6 +/− R/S	2.30 −06
Convert from CE to CB.	B	9.39 00 h_{ib}
	R/S	−2.28 −04 h_{rb}
	R/S	−9.94 −01 h_{fb}
	R/S	1.27 −08 h_{ob}
The values in CB configuration are stored in memory registers 1 through 4, so you can now perform the conversion to CC.	C	1.70 03 h_{ie}
	R/S	1.00 00 h_{re}
	R/S	−1.81 02 h_{fe}
	R/S	2.30 −06 h_{oe}

If you press A now, you'll get the CE parameters again.

References/ Information

J. A. Walston and J. R. Miller, *Transistor Circuit Design* (New York: McGraw-Hill Book Co., 1963), chapter 6.

Bode/Nyquist Calculations

OBJECTIVE

Given a transfer function or immittance function as a ratio of polynomials, this program calculates the data needed for Bode and Nyquist plots for specified frequency values.

FACTS/ THEORY

Given the transfer function or immittance function

$$H(s)=\frac{b_0+b_1s+\cdots+b_ns^m}{a_0+a_1s+\cdots+a_ns^n} \qquad m\le n\le 7$$

the program calculates the Bode-plot data

$$H_{dB}=20\log_{10}|H(j\omega)|$$
$$\measuredangle H=\arg\left[H(j\omega)\right] \quad \text{(degrees)}$$

and the rectangular and polar Nyquist-plot data

$$H_r=\mathrm{Re}\left[H(j\omega)\right] \qquad |H|=|H(j\omega)|$$
$$H_i=\mathrm{Im}\left[H(j\omega)\right] \qquad \measuredangle H=\arg\left[H(j\omega)\right] \quad \text{(degrees)}$$

Data may be calculated at a single frequency (ω) or at multiple frequencies ($\omega=\omega_0+n\Delta\omega$). The method used is to substitute $j\omega$ in H(s), giving

$$H(j\omega)=\frac{b_0+b_1(j\omega)+\ldots b_n(j\omega)^m}{a_0+a_1(j\omega)+\ldots a_n(j\omega)^n} \qquad m\le n\le 7$$

The real and imaginary parts of the numerator are

$$N_r=b_0-b_2\omega^2+b_4\omega^4-b_6\omega^6$$
$$N_i=\omega(b_1-b_3\omega^2+b_5\omega^4-b_7\omega^6)$$

For the denominator the real and imaginary parts are

$$D_r=a_0-a_2\omega^2+a_4\omega^4-a_6\omega^6$$
$$D_i=\omega\,(a_1-a_3\omega^2+a_5\omega^4-a_7\omega^6)$$

By substitution

$$H(j\omega)=\frac{N_r+jN_i}{D_r+jD_i}$$

where $b_7=0$ if $m<7$, etc.

Multiplying by the complex conjugate.

$$H(j\omega)=\frac{N_r+jN_i}{D_r+jD_i}\left[\frac{D_r-jD_i}{D_r-jD_i}\right]$$

gives

$$H_r=\frac{D_rN_r+D_iN_i}{D_r^2+D_i^2} \quad \text{and} \quad H_i=\frac{D_rN_i-D_iN_r}{D_r^2+D_i^2}$$

10

Bode/Nyquist Calculations

By rectangular to polar conversion $|H|$ and $\measuredangle H$ may be calculated.

This program also applies to frequency response functions of the form

$$H(f) = \frac{b_0 + b_1(jf) + \ldots b_m(jf)^m}{a_0 + a_1(jf) + \ldots a_n(jf)^n}$$

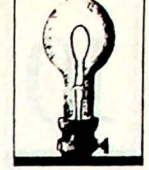
simply by entering f or f_0 instead of ω. Note the similarity in the four equations for N_r, N_i, D_r, and D_i. They all have the form

$$c - d\omega^2 + e\omega^4 - f\omega^6$$

where c, d, e, and f are some constants. Since the coefficients of these four equations are stored sequentially, the program will use indirect memory addressing and a subroutine to calculate N_r, N_i, D_r, and D_i.

The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enter learn mode.
000	2nd Lbl 2nd E' 18 STO 9 (RCL 2nd Ind 8 − E + E − E) INV SBR	Calculates equations in the form of $c - d\omega^2 + e\omega^4 - f\omega^6$. Recalls a's or b's indirectly from memory 8.
017	2nd Lbl E 2nd Op 28 2nd Op 28 2nd Op 29 (RCL 2nd Ind 8 × RCL 2nd Ind 9) INV SBR	Increments memory 8 by 2 to recall even- or odd-numbered coefficients. Increments memory 9 by 1 to recall proper power of ω.
033	2nd Lbl A 2nd CMS 8 STO 8 2nd Deg R/S	Initializes memories for indirect input of a_i values.
041	2nd Lbl 2nd A' STO 2nd Ind 9 2nd Op 29 R/S 2nd Dsz 8 2nd A' 49 +/− $\sqrt{x}$ R/S	Stores values of a_i sequentially, beginning in memory 0 (a_0 stored in memory 0, a_1 stored in memory 1, etc.). Memory 8 used for loop to prevent entry of more than seven coefficients.



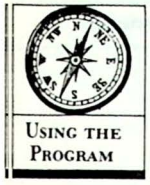
Step #	Program Keystrokes	Purpose/Comments
056	2nd Lbl B 8 STO 08 10 STO 9 R/S	Initializes memories for indirect storage of b_i values, and then stops for entry of first value.
066	GTO 2nd A'	Goes to subroutine A' for indirect storage of b's.
068	2nd Lbl C STO 18 x^2 STO 19 × x^2 STO 20 = STO 21	Begins single-frequency calculations. Stores ω; then calculates and stores ω^2, ω^4, and ω^6.
082	CLR STO 8 2nd E' STO 22 x^2 STO 26	Initiates memory 8 and calculates D_r. D_r and D_r^2 stored for later use.
091	10 STO 8 2nd E' STO 23	Initializes memory 8; calculates and stores N_r.
098	1 STO 8 2nd E' × RCL 18 = STO 24 x^2 SUM 26	Initializes memory 8; then calculates and stores D_i and $D_i^2 + D_r^2$.
111	11 STO 8 2nd E' × RCL 18 = STO 25	Initializes memory 8 to calculate and store N_i.
122	RCL 22 × RCL 23 + RCL 24 × RCL 25 = ÷ RCL 26 = STO 27	Calculates and stores $H_r = \dfrac{D_r N_r + D_i N_i}{D_r^2 + D_i^2}$
140	RCL 22 × RCL 25 − RCL 24 × RCL 23 = ÷ RCL 26 = STO 23	Calculates and stores $H_i = \dfrac{D_r N_i - D_i N_r}{D_r^2 + D_i^2}$
158	RCL 18 2nd Prt R/S	Displays ω and stops.
162	RCL 27 $x{:}t$ RCL 23 INV 2nd P→R 2nd Prt R/S	*Bode-plot data* Displays $\angle$ H and stops.
171	$x{:}t$ 2nd log × 20 = 2nd Prt R/S	Displays H_{dB} and stops.

Step #	Program Keystrokes	Purpose/Comments
179	2nd Lbl D RCL 27 2nd Prt R/S	*Nyquist-plot data* Displays H_r and stops.
185	x⇄t RCL 23 2nd Prt R/S	Displays H_i and stops.
190	INV 2nd P→R 2nd Prt R/S	Displays $\angle H$ and stops.
194	x⇄t 2nd Prt 2nd Adv R/S	Displays $\|H\|$ and stops.
198	RCL 29 SUM 18 RCL 18 GTO C	Increments ω and $\Delta\omega$ (for multiple-frequency calculations) and transfers to C to generate new values.
206	2nd Lbl 2nd C' STO 29 R/S GTO C	Begins multiple-frequency calculation. Stores $\Delta\omega$; then transfers to C for storage of ω_0 and calculation of data.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS

$R_{00} - R_{07}$	a_i values, as needed	R_{21}	ω^6
R_{08} }	looping and indirect	R_{22}	D_r
R_{09} }	memory addressing	R_{23}	N_r, H_i
$R_{10} - R_{17}$	b_i values, as needed	R_{24}	D_i
R_{18}	ω	R_{25}	N_i
R_{19}	ω^2	R_{26}	$D_r^2 + D_i^2$
R_{20}	ω^4	R_{27}	H_r
		R_{28}	
		R_{29}	$\Delta\omega$



Procedure	Press	Display/Comments
Set decimal for desired accuracy.	2nd fix n	
Initialize and enter *a* values.	A	8.
	a_0 R/S	a_0
	a_1 R/S	a_1
	•••	•••
Continue until all *a* values are entered.	a_n R/S	a_n If N>7, the display will show a flashing 7 to indicate that polynomials above the 7th order can't be handled.
Initialize and enter *b* values.	B	10.
	b_0 R/S	b_0
	b_1 R/S	b_1
	•••	•••
Continue until all *b* values are entered.	b_n R/S	b_n or flashing 7 if n>7.
For single-frequency calculations:		
Enter ω.	ω C	ω°
Generate data.	R/S	∠ H° } Bode data.
	R/S	H_{dB}° } Bode data.
	R/S or D	H_r° } Nyquist data.
	R/S	H_i° } Nyquist data.
	R/S	∠ H° } Nyquist data.
	R/S	|H|° } Nyquist data.

Procedure	Press	Display/Comments
For multiple-frequency calculations:		
Enter Δω.	Δω 2nd C'	Δω
Enter ω.	ω R/S	ω°
Calculate data.	R/S	∠ H° } Bode data.
	R/S	H_{dB}° } Bode data.
	R/S or D	H_r° } Nyquist data.
	R/S	H_i° } Nyquist data.
	R/S	∠ H° } Nyquist data.
	R/S	|H|° } Nyquist data.

°This value is printed if the PC-100A is used.

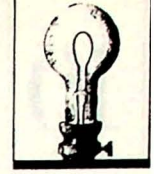
SAMPLE PROBLEM

Calculate the Bode- and Nyquist-plot data for

$$H(s) = \frac{100}{100 + 5s + s^2}$$

at ω = 8, 10, . . .

Procedure	Press	Display/Comments
Enter program.		0
Set decimal, if desired.	2nd Fix 2	0.00 We'll use 2-place decimal for convenience.
Initialize and enter *a* values.	A	8.00 (Used for Dsz loop.)
	100 R/S	100.00 a_0 (Stored in memory 0.)
	5 R/S	5.00 a_1 (Stored in memory 1.)
	1 R/S	1.00 a_2 (Stored in memory 2.)
Initialize and enter *b* values. (There is only one *b* value in this example.)	B	10.00 (Memory location for b_0.)
	100 R/S	100.00 b_0 (Stored in memory 10.)
Enter Δω and select multiple-frequency calculation.	2 2nd C'	2.00

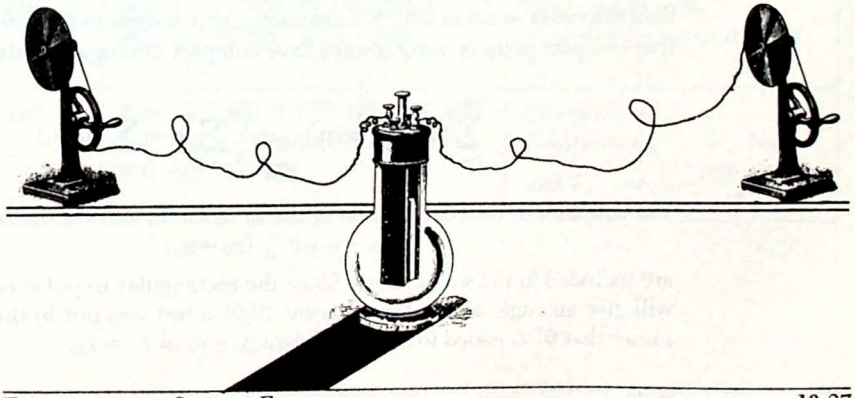


Procedure	Press	Display/Comments
Enter ω_0.	8 R/S	8.00 ω_0
	R/S	−48.01 ∠ H } Bode data.
	R/S	5.38 H_{dB} } Bode data.
	R/S	1.24 H_r } Nyquist data.
	R/S	−1.38 H_i } Nyquist data.
	R/S	−48.01 ∠ H } Nyquist data.
	R/S	1.86 |H| } Nyquist data.
	R/S	10.00 ω_1
	R/S	−90.00 ∠ H
	R/S	6.02 H_{dB}
	R/S	0.00 H_r
	R/S	−2.00 H_i
	R/S	−90.00 ∠ H
	R/S	2.00 |H|
	R/S	12.00 ω_2
	•••	•••

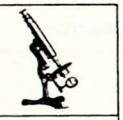
Continue to press R/S until you have generated all the data sets you need.

REFERENCES/ INFORMATION

Dean K. Frederick and A. Bruce Carlson, *Linear Systems in Communication and Control* (New York: John Wiley and Sons, Inc., 1971), chapter 11.



10 Root-Locus Calculations



OBJECTIVE

Given the open-loop poles and zeros of a linear feedback system, this program calculates the following root-locus parameters: asymptote intersection point, asymptote angles, departure angles from complex poles, and arrival angles at complex zeros.

FACTS/ THEORY

Given
$$F(s) = \frac{\prod_{i=1}^{m} (s - z_i)}{\prod_{k=1}^{n} (s - p_k)} \qquad m \leq n - 1$$

where z_i are the zeros and p_k the poles of F (s), the root-locus has $n - m$ asymptotes intersecting at

$$\sigma_0 = \frac{1}{n-m} \left(\sum_{k=1}^{n} p_k - \sum_{i=1}^{m} z_i \right)$$

at the angles
$$\psi_\nu = \frac{1}{n-m} (180° + (\nu - 1)\, 360°) \qquad \nu = 1, 2, \ldots, n - m$$

The departure angle at any complex pole p_J is

$$\Phi_J = \sum_{i=1}^{m} \measuredangle (p_J - z_i) - \sum_{\substack{k=1 \\ k \neq J}}^{n} \measuredangle (p_J - p_k) + 180° \qquad J = 1, 2, \ldots, n$$

The arrival angle at any complex zero z_K is

$$\Theta_K = - \sum_{\substack{i=1 \\ i \neq K}}^{m} \measuredangle (z_K - z_i) + \sum_{k=1}^{n} \measuredangle (z_K - p_k) - 180° \qquad K = 1, 2, \ldots, m$$

The following program computes σ_0, ψ_ν, Φ_J, and Θ_K for systems no greater than 6th order ($n \leq 6, m \leq 5$). We assumed a real system in the program so that complex poles or zeros always have complex-conjugate mates. Thus

$$\sum_{k=1}^{n} p_k = \sum_{k=1}^{n} \mathrm{Re}(p_k) \quad \text{and} \quad \sum_{i=1}^{m} z_i = \sum_{i=1}^{m} \mathrm{Re}(z_i)$$

You will note in the computation of the sums for Φ_J and Θ_K the terms

$$\measuredangle (p_J - p_J) \text{ or } \measuredangle (z_K - z_K)$$

are included in the summations. Since the rectangular to polar conversion will give an angle of 45° for the point, (0,0), a test was put in the program to insure that 0° is added to the sums when $p_k = p_J$ or $z_i = z_K$.

THE PROGRAM

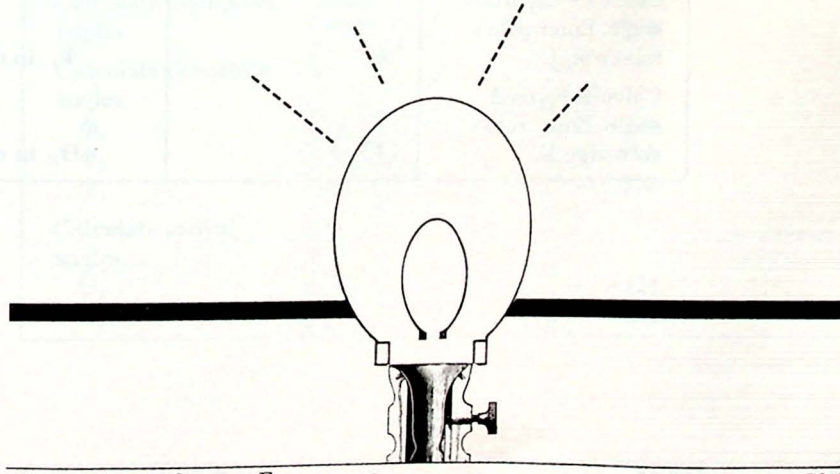
Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 04 5 STO 01 11 STO 2 RCL 4 R/S	Stores n. Stores starting memory locations for storage of p's. Recalls n and stops.
014	2nd Lbl 2nd A' STO 2nd Ind 1 R/S STO 2nd Ind 2 2nd Op 21 2nd Op 22 R/S GTO 2nd A'	Stores $\mathrm{Re}(p_k)$ and $\mathrm{Im}(p_k)$ for $k=1, 2, \ldots, n$.
028	2nd Lbl B STO 17 18 STO 01 24 STO 2 RCL 17 R/S	Stores m. Stores starting memory locations for storage of z's. Recalls m and stops.
043	2nd Lbl 2nd B' STO 2nd Ind 1 R/S STO 2nd Ind 2 2nd Op 21 2nd Op 22 R/S GTO 2nd B'	Stores Re (z_i) and Im (z_i) for $i=1, 2, \ldots, m$.
057	2nd Lbl C CLR STO 3 INV 2nd Fix RCL 4 STO 00 5 STO 01 18 STO 02	Calculates intersection point, σ_0. Initializes registers for loops.
075	2nd Lbl 2nd C' RCL 2nd Ind 1 − RCL 2nd Ind 2 = SUM 3 2nd Op 21 2nd Op 22 2nd Dsz 0 2nd C'	Computes $\sum_{i=1}^{n} \mathrm{Re}\,(p_i) - \mathrm{Re}\,(z_i)$. Note that $\mathrm{Re}\,(z_n)=0$.
092	RCL 4 − RCL 17 = STO 0 RCL 3 ÷ RCL 0 = R/S	Stores $n-m$ in R_0. Completes computation of σ_0.
107	180 ÷ RCL 0 = R/S 2nd Lbl 2nd Deg + 360 ÷ RCL 0 = R/S GTO 2nd Deg	Computes ψ_1. Computes $\psi_2, \ldots \psi_m$. Note that $\psi_{n+1} = \psi_n + \frac{360}{n-m}$.

Step #	Program Keystrokes	Purpose/Comments
128	2nd Lbl D + 4 = STO 23 + 6 = STO 29 0 STO 3 2nd Fix 0 RCL 4 STO 00 5 SBR SUM	Computes Φ_J or Θ_K. Stores register number for Re (p_J) and Im (p_J) or Re (z_K) and Im (z_K). Calls subroutine SUM to compute $\sum_{k=1}^{n} \angle(p_J - p_k)$ or $\sum_{k=1}^{n} \angle(z_K - p_K)$.
152	1 +/- 2nd Prd 3 RCL 17 STO 00 18 SBR SUM	Multiplies sums by -1. Call subroutine SUM to add sums $\sum_{i=1}^{m} \angle(p_J - z_i)$ and $\sum_{i=1}^{m} \angle(z_k - z_i)$.
164	RCL 3 + 180 = INV SBR	Adds 180 to final sums. If computing Φ_J, displays Φ_J and stops.
172	2nd Lbl E + 13 SBR D +/- R/S	Calls subroutine D to compute $-\Theta_K$. Displays Θ and stops.
181	2nd Lbl SUM STO 1 + 6 = STO 02	Stores starting register numbers for p_k or z_i.
190	2nd Lbl 2nd Rad 2nd CP INV 2nd St flg 1 RCL 2nd Ind 23 − RCL 2nd Ind 1 = x:t INV 2nd x=t 209 2nd St flg 1	Clears t register and unsets flag 1. Computes the real component (x-value) of $(p_J - p_k)$ or $(z_K - z_i)$ and puts it in the t register. If $x = 0$, sets flag 1. If $x \neq 0$, transfers to 209.

Step #	Program Keystrokes	Purpose/Comments
209	RCL 2nd Ind 29 − RCL 2nd Ind 2 = INV 2nd If flg. 1 223 2nd x=t 225	Computes the imaginary component (y-value). If flag 1 not set ($x \neq 0$) transfers to 223. If flag 1 set ($x = 0$), tests to see if $y = 0$. If $y = 0$, transfers to 225.
223	INV 2nd P→R	If x and y are both zero, adds zero to sum in register 3. If x and y are not both zero, does rectangular to polar conversions and adds the angle to sum in R_3. Loops n or m times, as needed.
225	SUM 3 2nd Op 21 2nd Op 22 2nd Dsz 0 2nd Rad INV SBR	
	LRN RST	Exits learn mode.

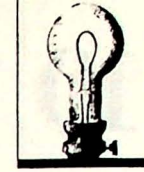
REGISTER CONTENTS

R_{00}	Used	R_{17}	m
R_{01}	Used	$R_{18} - R_{22}$	Re (z_i)
R_{02}	Used	R_{23}	Used
R_{03}	Used	$R_{24} - R_{28}$	Im (z_i)
R_{04}	n	R_{29}	Used
$R_{05} - R_{10}$	Re (p_k)		
$R_{11} - R_{16}$	Im (p_k)		



USING THE PROGRAM

Procedure	Press	Display/Comments
Enter the number of poles, n.	A	n
Enter p_k.		
Re (p_1)	R/S	Re (p_1)
Im (p_1)	R/S	Im(p_1)
• • •	• • •	• • •
Re (p_n)	R/S	Re (p_n)
Im (p_n)	R/S	Im (p_n)
Enter the number of complex zeros, m.	B	m
Enter z_i.		
Re (z_1)	R/S	Re (z_1)
Im (z_1)	R/S	Im (z_1)
• • •	• • •	• • •
Re (z_m)	R/S	Re (z_m)
Im (z_m)	R/S	Im (z_m)
Calculate asymptote intersection point.	C	σ_o
Calculate asymptote angles.	R/S	ψ_1 in degrees
	R/S	ψ_2 in degrees
	• • •	• • •
	R/S	ψ_{n-m} in degrees
Calculate departure angle. Enter pole's subscript, J.	D	Φ_J in degrees
Calculate arrival angle. Enter zero's subscript, K.	E	Θ_K in degrees



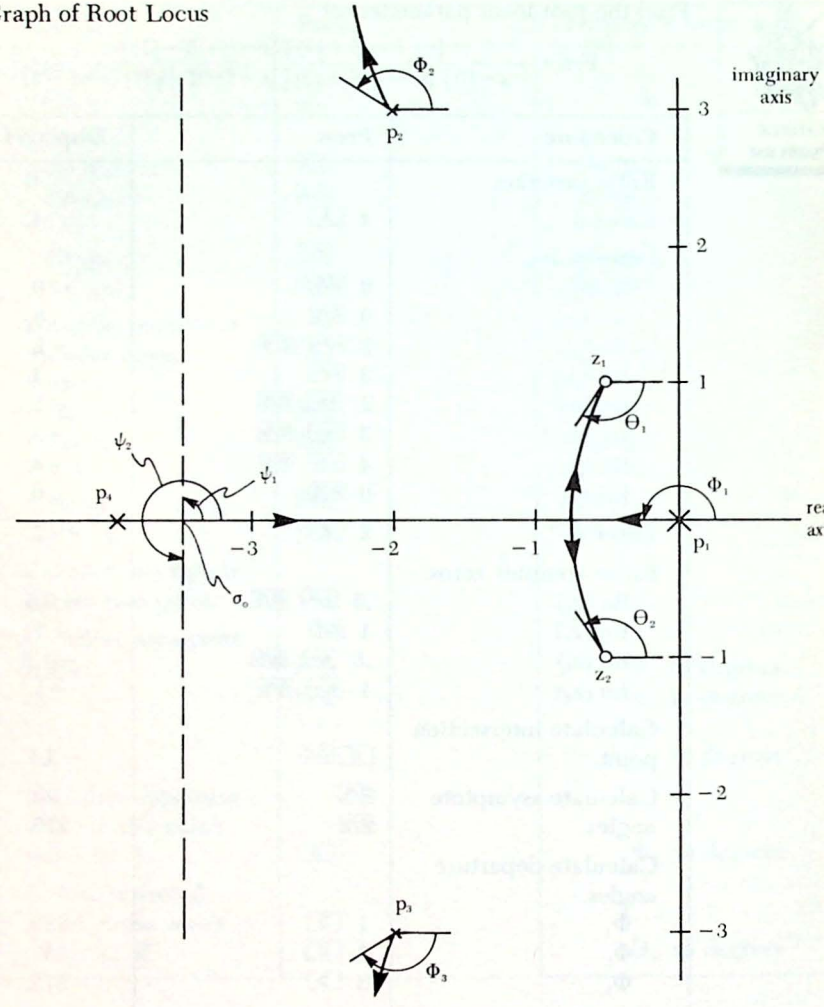
Sample Problem

Find the root-locus parameters of

$$F(s)=\frac{[s-(-.5+j1)]\,[s-(-.5-j1)]}{[(s-0)]\,[s-(-2+j3)]\,[s-(-2-j3)]\,[s-(-4)]}$$

Procedure	Press	Display/Comments
Enter program.		0
Enter n.	4 [A]	4.
Enter poles.		
Re (p_1)	0 [R/S]	0.
Im (p_1)	0 [R/S]	0.
Re (p_2)	2 [+/−] [R/S]	−2.
Im (p_2)	3 [R/S]	3.
Re (p_3)	2 [+/−] [R/S]	−2.
Im (p_3)	3 [+/−] [R/S]	−3.
Re (p_4)	4 [+/−] [R/S]	−4.
Im (p_4)	0 [R/S]	0.
Enter m.	2 [B]	2.
Enter complex zeros.		
Re (z_1)	.5 [+/−] [R/S]	−0.5
Im (z_1)	1 [R/S]	1.
Re (z_2)	.5 [+/−] [R/S]	−0.5
Im (z_2)	1 [+/−] [R/S]	−1.
Calculate intersection point.	[C]	−3.5
Calculate asymptote angles.	[R/S]	90.
	[R/S]	270.
Calculate departure angles.		
Φ_1	1 [D]	180.
Φ_2	2 [D]	147.
Φ_3	3 [D]	573.
Calculate arrival angles.		
Θ_1	1 [E]	−121.
Θ_2	2 [E]	121.

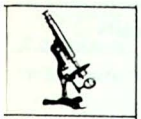
Graph of Root Locus



REFERENCES/ INFORMATION

D. K. Frederick and A. B. Carlson, *Linear Systems in Communication and Control* (New York: John Wiley and Sons, Inc., 1971), chapter 10.

Step and Pulse Response



OBJECTIVE

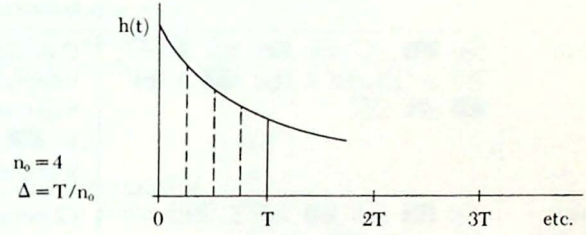
FACTS/ THEORY

If you know the impulse response h(t) of a linear system, it is possible to find the unit step response and the rectangular pulse response. The following program gives the value for the time interval (T) and the step or rectangular pulse response, whichever you select.

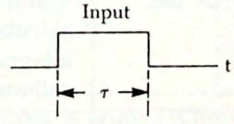
The unit step response is given by

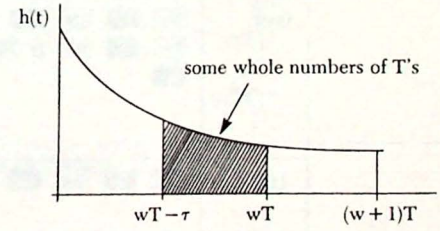
$$y(t)=\int_0^t h(\lambda)d\lambda$$

The integral will be evaluated at t = T, 2T, 3T, etc., using Master Library program ML-09 (Simpson's Approximation) to approximate the integral. You'll choose an even integer n_0 (the number of divisions from 0 to T that the approximation will use). The time between divisions is $T/n_0=\Delta$, and the program increases $n=t/\Delta$ to keep Δ constant as t increases.



The rectangular pulse response is found in the same way as the unit step response, until $t=\tau$ is reached. At that point the lower limit becomes $wT-\tau$ and the upper limit wT. The next value of the integral is from $wT-\tau+T$ to $(w+1)T$, etc.



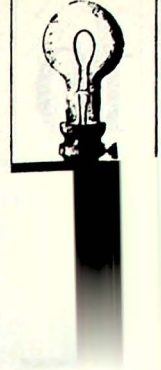


Before running the program, you must enter a subroutine to evaluate h(t) as a function of t. To test the numerical accuracy, you should recompute at least one point with the value of n_0 doubled.

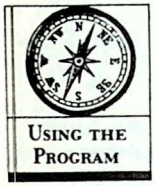
THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl 2nd E' 2nd Pgm 9 D RCL 2 2nd Prt R/S RCL 4 2nd Prt R/S RCL 0 SUM 2 INV SBR	Computes integral with Simpson's Rule.
018	2nd Lbl A STO 2 STO 0 CLR STO 1 R/S 2nd Pgm 9 C R/S GTO C	Stores T and n_0; then starts integration for unit step response.
034	2nd Lbl B STO 7 RCL 5 STO 6 2nd St flg 0 GTO 2nd B'	Enters τ for rectangular pulse response and sets flag 0 for testing later.
046	2nd Lbl C 2nd E' RCL 2 ÷ RCL 3 = STO 5 2nd If flg 0 2nd B' GTO C	Calculates new h for next integration; checks to see if flag 0 is set. If so, transfers to B' . If not, repeats integration.
062	2nd Lbl 2nd B' RCL 2 x:t RCL 7 2nd x≥t C	Checks to see if $\tau \geq$ t. If so, transfers to C . If not, continues on for new integration.
071	2nd Lbl 2nd C' RCL 2 − RCL 7 = STO 1 RCL 7 ÷ RCL 6 = STO 03	Stores $t-\tau$ as new lower limit; then calculates and stores new $\Delta = \tau/n_0$.
089	2nd Lbl 2nd D' RCL 6 STO 5 2nd E' RCL 0 SUM 1 GTO 2nd D'	Stores original n in memory 5, and then loops to calculate mT and value of integral from $wT-\tau$ to wT. Adjusts data for next run.
102	2nd Lbl 2nd A'	Labels subroutine for calculation of h(t).
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.



REGISTER CONTENTS				
	R_{00}	T	R_{05}	n
	R_{01}	t_0	R_{06}	n_0
	R_{02}	t	R_{07}	τ
	R_{03}	Δ	R_{08}	
	R_{04}	y	R_{09}	



USING THE PROGRAM

Procedure	Press	Display/Comments
Enter subroutine to calculate h(t). Assume that t is in the display, and enter keystrokes to evaluate function.	GTO 2nd A′ LRN keystrokes for h(t) calculation.	104 00
End subroutine with return command and exit learn mode.	INV SBR LRN	0
Enter T.	A	0
Enter n_0.	R/S	$\Delta = T/n_0$
For unit step response, proceed by pressing R/S	R/S R/S R/S ...	T° y(t)° 2T° ...
Continue to press R/S for new values.		
OR		
For rectangular pulse response, enter τ.	B R/S R/S R/S ...	T° y(T)° 2T° y(2T)° ...

°This value is printed if the PC-100A is used.

10 Step and Pulse Response

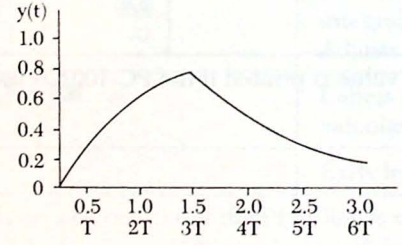
Sample Problem

Given $h(t) = e^{-t}$, find the pulse response y(t) for $\tau = 1.5$ with $n_0 = 4$ and $T = 0.5$.

Procedure	Press	Display/Comments
Enter program.		**0**
Enter h(t) subroutine.	GTO 2nd A LRN +/− INV ln x INV SBR LRN	**104 00** **0.**
Fix decimal for desired accuracy.	2nd fix 3	**0.000** (We'll use 3-place decimal for convenience.)
Enter T.	.5 A	**0.000**
Enter n_0.	4 R/S	**0.125** $\Delta = T/n_0$.
Enter τ.	1.5 B	**0.500** T—note that calculation takes several seconds.
	R/S	**0.393** y(T).
	R/S	**1.000** 2T
	R/S	**0.632** y(2T)
	R/S	**1.500** 3T
	R/S	**0.777** y(3T)
	R/S	**2.000** 4T
	R/S	**0.471** y(4T)
	R/S	**2.500** 5T
	R/S	**0.286** y(5T)
	R/S	**3.000** 6T
	R/S	**0.173** y(6T)

Continue until you have generated the desired number of values.

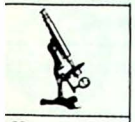
Graph of Results



References/ Information

D. K. Frederick and A. B. Carlson, *Linear Systems in Communication and Control* (New York: John Wiley and Sons, Inc., 1971), chapter 4.

Convolution



OBJECTIVE

FACTS/ THEORY

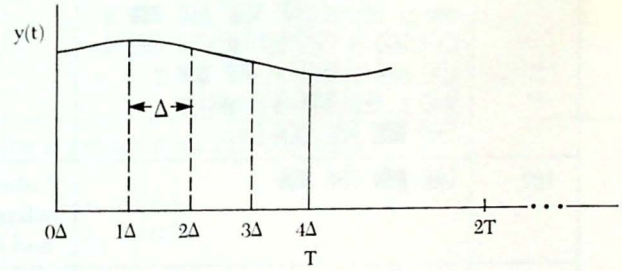
Given the impulse response for a linear system, you can use the convolution integral to find the system's output for an input waveform. The program shown here generates outputs at intervals of T, using the trapezoidal rule.

To calculate the output y(t) of a linear system, the convolution integral integrates the product of the input x(t) and the impulse response h(t) in the following form.

$$y(t) = \int_0^t x(\lambda)h(t-\lambda)d\lambda$$

Applying the trapezoidal rule to the convolution integral yields

$$y(t) \cong \Delta\left[\frac{x(0)h(t)}{2} + \sum_{k=1}^{n-1} x(k\Delta)h(t-k\Delta) + \frac{x(t)h(0)}{2}\right]$$



The number of divisions from 0 to T is n_o, and $\Delta = T/n_o$. Note that n is increased for each T value (2T, 3T, etc.) to keep Δ constant.

The program allows you to enter the special subroutines needed to calculate x(t) and h(t).

Convolution

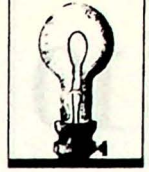
The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enter learn mode.
000	2nd Lbl A 2nd CMs STO 2 STO 3 CLR R/S 2nd Int STO 1 INV 2nd Prd 03	Stores T and n_0; then calculates and stores $T/n_0 = \Delta t$.
015	2nd Lbl B CLR STO 6 STO 7 RCL 2 SUM 5 RCL 1 SUM 4 RCL 4 STO 0 2nd Up 30	Begins calculation of y(T), y(2T), etc.
036	2nd Lbl 2nd C' RCL 3 SUM 6 RCL 6 2nd A' STO 8 RCL 5 − RCL 6 = 2nd B' × RCL 8 = SUM 7 2nd Dsz 0 2nd C' CLR 2nd A' STO 8 RCL 5 2nd B' 2nd Prd 8 CLR 2nd B' STO 9 RCL 5 2nd A' 2nd Prd 9 (RCL 8 + RCL 9) ÷ 2 = SUM 7 RCL 3 2nd Prd 7 RCL 5 2nd Prt R/S RCL 7 2nd Prt R/S GTO B	Begins loop to generate kΔ as 1Δ, 2Δ, . . . (n − 1)Δ. Calculates and displays successive values ot T_i and $y(T_i)$.
107	2nd Lbl 2nd A'	Labels point in program where subroutines for x(t) and h(t) will be entered.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

Register Contents

R_{00}	counter for loops	R_{04}	n	R_{08}	Used
R_{01}	n_0	R_{05}	t	R_{09}	Used
R_{02}	T	R_{06}	kΔ	R_{10}	
R_{03}	Δ	R_{07}	y	R_{11}	



ING THE ROGRAM

Procedure	Press	Display/Comments
Enter subroutine for x(t) and h(t).	GTO 2nd A'	
	LRN	109 00
	Keystrokes for x(t).	
	INV SBR	
	2nd tbl 2nd B'	
Be sure to end each subroutine with INV SBR.	Keystrokes for h(t)	
	INV SBR	
Exit learn mode.	LRN	0
Enter T.	A	0
Enter n_0.	R/S	T°
	R/S	y(T)°
	R/S	2T°
	R/S	y(2T)°
	R/S	3T°
	R/S	y(3T)°
	•••	•••

°This value is printed if the PC-100A is used.

SAMPLE ROBLEM

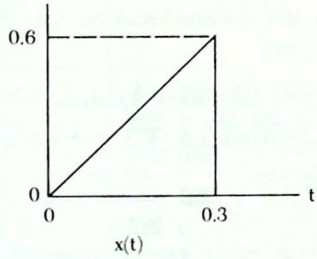
Given $x(t) = \begin{cases} 2t & t \leq 0.3 \\ 0 & t > 0.3 \end{cases}$

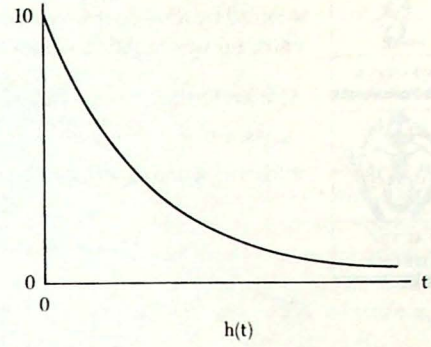
and $h(t) = 10e^{-5t}$,

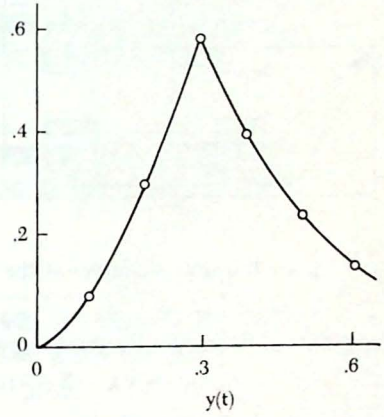
find y(t) at $t = 0.1, 0.2, \ldots$, using $n_0 = 4$.

Procedure	Press	Display/Comments
Enter program.		0
Enter subroutine for x(t).	GTO 2nd A'	
	LRN	109 00
Assume t is in display.	× 2 = x:t	
	.6 2nd x≥t	
	C 2nd CP 2nd	
	Lbl C	
	x:t	
End with INV SBR and enter subroutine for h(t).	INV SBR	122 00
	2nd Lbl 2nd	
	B' × 5 =	
	+/− INV ln x	
	× 10 =	
Be sure to end with return command.	INV SBR	135 00
Exit learn mode.	LRN	0.
Fix decimal, if desired.	2nd fix 3	0.000 We'll use 3-place decimal.
Enter T.	.1 A	0
Enter n_o.	4 R/S	0.100 T (allow several seconds for calculation of each successive T value).
	R/S	0.086 y(T)
	R/S	0.200 2T
	R/S	0.296 y(2T)
	R/S	0.300 3T
	R/S	0.581 y(3T)
	R/S	0.400 4T
	R/S	0.398 y(4T)
	R/S	0.500 5T
	R/S	0.241 y(5T)
	R/S	0.600 6T
	R/S	0.146 y(6T)

The problem may be graphed as follows.

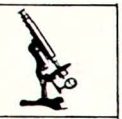






REFERENCES/ INFORMATION D. K. Frederick and A. B. Carlson, *Linear Systems in Communication and Control* (New York: John Wiley and Sons, Inc., 1971), chapter 4.

10 Discrete-Time System Response



OBJECTIVE

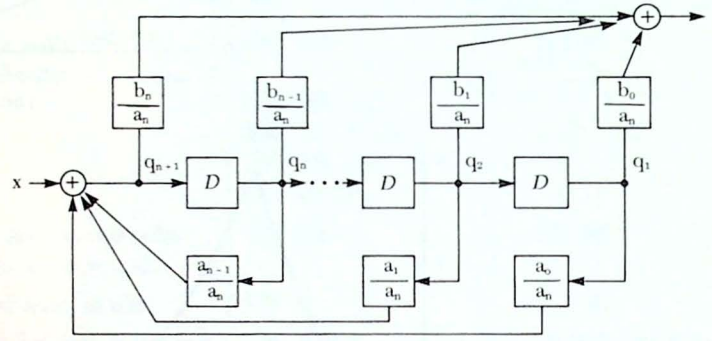
This program calculates output values for a discrete-time system described by a linear difference equation. The input may be entered manually or entered by a subroutine which computes the input values. Also, you may enter initial-condition values if you wish.

FACTS/ THEORY

A discrete-time system described by the difference equation

$$a_n y(k+n) + \cdots + a_1 y(k+1) + a_0 y(k) = b_m x(k+m) + \cdots + b_1 x(k+1) + b_0 x(k)$$

with $m \leq n$ and $a_n \neq 0$, can be modeled as:



where $q_1, q_2, \ldots q_n$ are the state variables at the outputs of the successive delayers *(D)*.

From the sum points of the model diagram, it follows that, for any k,

$$q_{n+1}(k) = x(k) - \sum_{i=0}^{n-1} \left(\frac{a_i}{a_n}\right) q_{i+1}$$

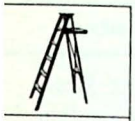
Hence

$$y(k) = \sum_{i=0}^{n} \left(\frac{b_i}{a_n}\right) q_{i+1}$$

and for the next iteration

$$q_i(k+1) = q_{i+1}(k) \qquad i = 1, 2, \cdots, n-1$$

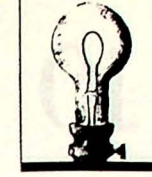
Given x(k), the following program computes successive values of y(k) for a system up to seventh order ($m \leq n \leq 7$). Initial values for $q_1, q_2, \ldots q_n$ may be entered. The input x(k) may be entered directly from the keyboard or computed in subroutine **E** as sample values of x(t). Note that sampling a signal every T seconds yields the sample values x(kT) for any k.



THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [2nd] [A'] [R/S] [STO] [2nd] [Ind] 1 [2nd] [Op] 21 [2nd] [Dsz] 0 [2nd] [A'] [INV] [SBR]	Stores values for a's, b's, and q's.
011	[2nd] [Lbl] [A] [2nd] [CMs] [INV] [2nd] [St flg] 0 [+] 1 [=] [STO] 4 [STO] 00 5 [STO] 1 [CLR] [2nd] [A'] [STO] 12 [R/S]	Clears memories and flag 0. Stores $n+1$ in R_4. Stores starting memory location for storing a_i values, Calls subroutine [A'] to store a_i values. Stores a_n in R_{12}.
032	[2nd] [Lbl] [B] [RCL] 4 [STO] 00 13 [STO] 1 [CLR] [2nd] [A'] [R/S]	Stores starting memory location for b_i values. Calls subroutine [A'] to store b_i values.
045	[2nd] [Lbl] [C] [RCL] 4 [STO] 0 [2nd] [Op] 30 21 [STO] 1 [CLR] [2nd] [A'] [R/S]	Stores n in R_0 for looping. Stores starting memory location for q_i values. Calls subroutine [A'] to store q_i values.
060	[2nd] [Lbl] [D] [STO] 29 [+/−] [STO] 3 [2nd] [St flg] 0 [GTO] [2nd] [B']	Stores T in R_{29}. Sets $kT = -T$ for first value. Sets flag 0 to indicate $x(kT)$ to be used. Transfers to [B'].
071	[2nd] [Lbl] [E] 1 [+/−] [STO] 03	Sets $k = -1$ for first value.
077	[2nd] [Lbl] [2nd] [B'] [RCL] 4 [STO] 00 5 [STO] 01 13 [STO] 02	Stores $n+1$ in R_0. Stores starting memory locations for a_i and b_i values.
090	[2nd] [Lbl] [2nd] [Prd] [RCL] 12 [INV] [2nd] [Prd] [2nd] [Ind] 1 [INV] [2nd] [Prd] [2nd] [Ind] 2 [2nd] [Op] 21 [2nd] [Op] 22 [2nd] [Dsz] 0 [2nd] [Prd]	Recalls a_n and divides it into $a_0, a_1, \ldots, a_n$ and $b_0, b_1, \ldots, b_n$.

Step #	Program Keystrokes	Purpose/Comments
107	2nd Lbl SUM 0 STO 12 5 STO 01 21 STO 2 RCL 4 STO 0 2nd Op 30 2nd D'	Clears memory 12. Stores starting memory location of $\frac{a_i}{a_n}$ and of q_i values. Calls subroutine D' to compute $\Sigma \frac{a_i}{a_n} q_{i+1}$.
126	RCL 12 +/− STO 2nd Ind 2 2nd If flg 0 2nd x=t	Stores $-\Sigma \frac{a_i}{a_n} q_{i+1}$ in R_{28}. If flag 0 is set, go to x=t.
134	2nd Op 23 RCL 3 2nd Prt R/S GTO 2nd x=t	If x(k) is being entered, add 1 to k and stop for x(k) to be entered.
142	2nd Lbl 2nd x=t RCL 29 SUM 3 RCL 3 2nd Prt R/S 2nd E'	Recalls T and places kT+T in R_3. Displays kT and calls subroutine E' to compute x(kT).
153	2nd Lbl 2nd x=t SUM 2nd Ind 02 0 STO 12 13 STO 01 21 STO 2 RCL 4 STO 0 2nd D'	Adds x to $-\Sigma \frac{a_i}{a_n} q_{i+1}$. Initializes registers and calls subroutine D' to calculate $\Sigma \frac{b_i}{a_n} q_{i+1}$.
173	RCL 12 2nd Prt R/S	Displays y(k).
177	21 STO 01 22 STO 2 RCL 4 STO 0 2nd Op 30	Advances q's for next calculation, i.e., moves q_2 to q_1 position, q_3 to q_2 position, etc.
191	2nd Lbl 2nd Adv RCL 2nd Ind 2 STO 2nd Ind 1 2nd Op 21 2nd Op 22 2nd Dsz 0 2nd Adv GTO SUM	
206	2nd Lbl 2nd D' RCL 2nd Ind 1 × RCL 2nd Ind 2 = SUM 12 2nd Op 21 2nd Op 22 2nd Dsz 0 2nd D' INV SBR	Computes $\Sigma \frac{a_i}{a_n} q_{i+1}$ or $\Sigma \frac{b_i}{a_n} q_{i+1}$. Stores sum in R_{12}.



Step #	Program Keystrokes	Purpose/Comments
224	2nd Lbl 2nd E'	Label for x(kT) subroutine.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS	R_{00}	Used	R_{03}	k or kT	$R_{13}-R_{20}$	b_i	R_{29}	T,x
	R_{01}	Used	R_{04}	n+1	$R_{21}-R_{27}$	q_i		
	R_{02}	Used	$R_{05}-R_{12}$	a_i	R_{28}	Used		

Procedure	Press	Display/Comments
Enter n.	A	0.
Enter a_i values.		
a_0	R/S	a_0
a_1	R/S	a_1
• • •	• • •	• • •
a_n	R/S	a_n
Enter b_i values.	B	0.
b_0	R/S	b_0
b_1	R/S	b_1
• • •	• • •	• • •
b_n	R/S	b_n
Enter initial q_i values if desired.	C	0.
q_1	R/S	q_1
q_2	R/S	q_2
• • •	• • •	• • •
q_n	R/S	q_n
If you are entering each x(k), follow this procedure.		
Begin calculation of y(0).	E	0.° k=0
Enter x(0).	1 R/S	y(0°
Begin calculation of y(1).	R/S	1.° k=1
Enter x(1).	1 R/S	y(1)°

Procedure	Press	Display/Comments
Begin calculation of y(2).	R/S	2.° k=2
Enter x(2).	1 R/S	y(2)°
Continue in this way until all desired values are generated.		
OR		
If you are using a subroutine to compute x(kT), use this procedure.		
Enter subroutine for x(kT).	CLR GTO 2nd [redacted] LRN Keystrokes for x(kT). INV SBR LRN	226 00 0.
Enter T value.	D	0.°
Calculate y(0).	R/S	y(0)°
Begin calculation of y(1).	R/S	1T°
Calculate y(1).	R/S	y(1)°
Begin calculation of y(2).	R/S	2T°
Calculate y(2).	R/S	y(2)°
Continue in this way until all desired values are generated.		

°This value will be printed if the PC-100A is used.

SAMPLE PROBLEM

(1) For the difference equation

$$3y(k+1)+1.5y(k) = -3x(k+1)+6x(k)$$

find y(k) when x(k) = 1 for k = 0, 1, 2, ... Use an initial value $q_1 = 2$.

Procedure	Press	Display/Comments
Enter program.		0
Enter n.	1 [A]	0
Enter a_i values.		
a_0	1.5 [R/S]	1.5
a_1	3 [R/S]	3.
Enter b_i values.	[B]	0
b_0	6 [R/S]	6.
b_1	3 [+/−] [R/S]	−3.
Enter initial q_i values.	[C]	0
q_1	2 [R/S]	2.
Begin calculation of y(0).	[E]	0. k
Enter x(0) and calculate y(0).	1 [R/S]	4. y(0)
Begin calculation of y(1).	[R/S]	1. k
Enter x(1) and calculate y(1).	1 [R/S]	−1. y(1)
Begin calculation of y(2).	[R/S]	2. k
Enter x(2) and calculate y(2).	1 [R/S]	1.5 y(2)
Continue in this way until all values desired are calculated.		

(2) Given the difference equation

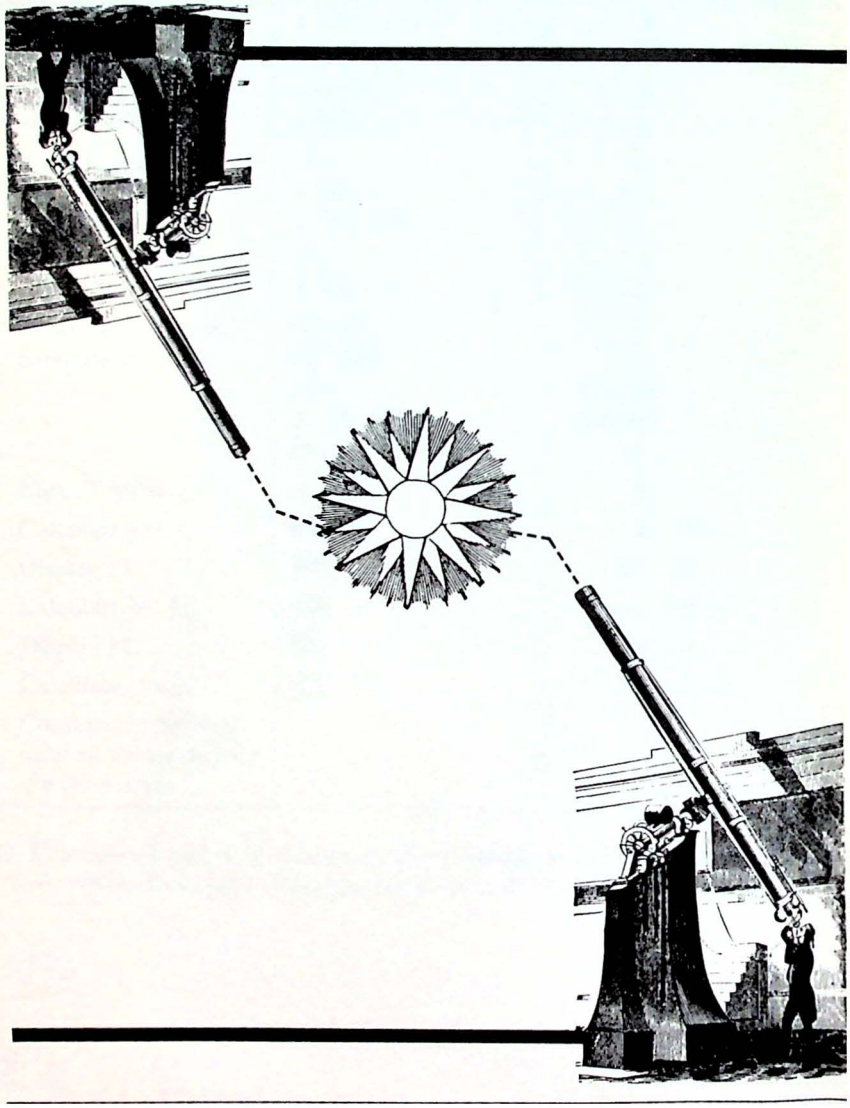
$$3y(kT+T)+1.5y(kT)=-3x(kT+T)+6x(kT)$$

find y(kT), T = 0.1, when x(t) = 1 for t ≥ 0. Use $q_1 = 2$ for an initial value.

Procedure	Press	Display/Comments
Enter n.	1 [A]	0
Enter a_i values.		
a_0	1.5 [R/S]	1.5
a_1	3 [R/S]	3.
Enter b_i values.	[B]	0
b_0	6 [R/S]	6.
b_1	3 [+/−] [R/S]	−3.
Enter q_i values.	[C]	0
q_1	2 [R/S]	2.
Enter subroutine to compute x(t).	[CLR] [GTO]	
	[2nd] [E']	0.
	[LRN]	226 00
	1 [INV] [SBR]	228 00
	[LRN]	0.
Enter T value.	.1 [D]	0.
Calculate y(0).	[R/S]	4. y(0)
Display 1T.	[R/S]	0.1 1T
Calculate y(0.1).	[R/S]	−1. y(0.1)
Display 2T.	[R/S]	0.2 2T
Calculate y(0.2).	[R/S]	1.5 y(0.2)
Continue in this way until all values desired are generated.		

REFERENCES/ INFORMATION D. K. Frederick and A. B. Carlson, *Linear Systems in Communication and Control* (New York: John Wiley and Sons, Inc., 1971), chapter 13.

11 Physics and Astronomy



Introduction

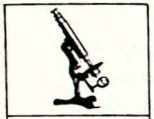
The handheld programmable calculator, by its very existence, opens several new and important avenues of approach for the student of physics and/or astronomy. In this chapter our aim is to present a selection of programming examples showing the calculator as a new exploratory tool for the scientist. We'll focus in particular on the role of the calculator in allowing you to:

- graph and analyze the behavior of an equation or mathematical model,
- apply the simple but powerful numerical integration techniques to situations involving differential equations, and
- set up a "numerical laboratory" where the simulation of a complete system can be programmed and investigated.

By no means do the examples presented here represent a comprehensive collection of calculator applications in physics and astronomy. They were chosen simply to illustrate some of the potential of the use of the programmable calculator in the field of the physical sciences. We hope that these examples will serve as a springboard for teachers, students, and professionals alike.

11

Fraunhofer Diffraction at a Circular Aperture



OBJECTIVE

An important diffraction situation in practice is the case of Fraunhofer diffraction at a circular aperture. The following program allows you to graph intensity vs. angle behavior for any selected values of the parameters R/λ and N. (Here R is the aperture radius, λ is the wavelength of the incident plane radiation, and N is the number of "strips" we divide the circular aperture into for purposes of analysis.)

With this program it's easy to explore the intensity behavior predicted as key parameters are varied. Note that performing the calculations involved, even with the aid of a standard "slide-rule" calculator, can be quite tedious and time consuming. The programmable calculator enables you to store an entire "mathematical model" in the machine so that you can focus your attention more directly on the nature of the predicted results, rather than on the computations involved in getting at them.

FACTS/ THEORY

The usual treatment of this problem in elementary physics courses and textbooks is first to study Fraunhofer diffraction at a long slit, since this problem is easier to tackle, and then to illustrate and state the result for a circular aperture. With the programmable calculator it's easy to make an accurate graph of the circular aperture diffraction pattern using a straightforward procedure completely analogous to the derivation of the single slit results. We'll make all the usual assumptions for Fraunhofer diffraction found in most elementary treatments.

As shown in *Figure 1*, we'll assume plane waves of wavelength λ perpendicularly incident on a circular aperture of radius R in an opaque screen.

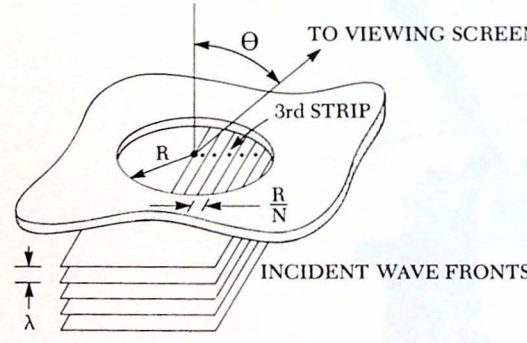


Figure 1.

We'll assume that everywhere in the aperture the scalar wave field is unaffected by the presence of the screen and that the pattern is viewed on an infinitely distant screen.

At a given angle Θ from the axis of symmetry, the wave amplitude on the viewing screen is calculated as follows. Half of the aperture is divided into N strips of equal width R/N, where N should be greater than 10 for reasonable accuracy. (See *Figure 1.*) The contribution E_i to the total wave amplitude E on the viewing screen, due to the i-th strip in the aperture, is approximated by

$$E_i = KA_i \cos[(i - \tfrac{1}{2})\phi] \qquad 1 \le i \le N.$$

In this formula K is a constant, A_i is the area of the i-th strip, and $(i - \frac{1}{2})\phi$ is the phase shift of the ray from the midpoint of the i-th strip, relative to the ray from the center of the aperture. ϕ is the phase difference between rays from the midpoints of neighboring strips and is given by

$$\phi = 2\pi \frac{R}{N\lambda} \sin\Theta$$

The area of the i-th strip can be calculated by means of direct integration:

$$A_i = R^2\left[\cos^{-1}\sqrt{1-(\frac{i}{N})^2} + \frac{i}{N}\sqrt{1-(\frac{i}{N})^2} - \cos^{-1}\sqrt{1-(\frac{i-1}{N})^2} - \frac{i-1}{N}\sqrt{1-(\frac{i-1}{N})^2}\right]$$

$$1 \le i \le N.$$

The total amplitude $E(\Theta)$ on the screen is then calculated from

$$E(\Theta) = 2K\sum_{i=1}^{N} A_i \cos[(i - \tfrac{1}{2})\phi]$$

where the factor of 2 accounts for the effect of the other half of the aperture. If we choose units such that $K = (\pi R^2)^{(-1)}$, then $E(0)$ has the value unity. Our program is arranged to compute and display $E(\Theta)^2$, which is proportional to the actual pattern intensity observable on the viewing screen, $I(\Theta)$.

11

Fraunhofer Diffraction at a Circular Aperture

The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [B] [2nd] [sin] [×] [RCL] 7 [=] [STO] 8 [CLR] [STO] 10 [STO] 11 [STO] 02 .5 [+/−] [STO] 1 [RCL] 6 [STO] 00	Enters Θ and computes ϕ.
025	[2nd] [Lbl] [2nd] [A'] 1 [SUM] 1 [RCL] 10 [STO] 9 [RCL] 12 [SUM] 2 [RCL] 2 [x^2] [+/−] [+] 1 [=] [$\sqrt{x}$] [STO] 13 [×] [RCL] 2 [=] [+] [RCL] 13 [INV] [2nd] [cos] [=] [STO] 10 [−] [RCL] 9 [=] [×] [(] [RCL] 8 [×] [RCL] 1 [)] [2nd] [cos] [=] [SUM] 11 [2nd] [Dsz] 0 [2nd] [A']	Computes $A_i \cos[(i-\frac{1}{2})\phi]$. Accumulates $\sum_{i=1}^{N} A_i \cos[(i-\frac{1}{2})\phi]$ in memory 11.
079	[RCL] 11 [×] 2 [÷] [2nd] [π] [=] [x^2] [2nd] [Prt] [2nd] [Adv] [INV] [SBR]	Computes $I(\Theta) = E(\Theta)^2$.
090	[2nd] [Lbl] [C] [STO] 14 [R/S] [STO] 15 [R/S] [STO] 16 [2nd] [Lbl] [E] [RCL] 14 [2nd] [Prt] [R/S] [B] [R/S] [RCL] 16 [SUM] 14 [RCL] 15 [−] [RCL] 14 [=] [2nd] [x≥t] [E] [R/S]	Stores Θ_0, Θ_f, and $\Delta\Theta$. Then calculates consecutive values of $I(\Theta)$ in steps of $\Delta\Theta$.
121	[2nd] [Lbl] [A] [STO] 5 [R/S] [STO] 6 [1/x] [STO] 12 [×] [RCL] 5 [×] 2 [×] [2nd] [π] [=] [STO] 7 [2nd] [Rad] [R/S]	Stores R/λ, N, and $2\pi R/(N\lambda)$.
	[LRN] [RST]	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

Register Contents

R_{00}	Used	R_{06}	N	R_{12}	$1/N$
R_{01}	$i-\frac{1}{2}$	R_{07}	$2\pi R/(\lambda N)$	R_{13}	$\sqrt{1-(\frac{i}{N})^2}$
R_{02}	i/N	R_{08}	ϕ	R_{14}	Θ_i
R_{03}	Used	R_{09}	$A_i - 1$	R_{15}	Θ_f
R_{04}		R_{10}	A_i	R_{16}	$\Delta\Theta$
R_{05}	Θ_i	R_{11}	$\sum_{K=1}^{i} E_K$	R_{17}	

USING THE PROGRAM

Procedure	Press	Display/Comments
Clear memories and set decimal for desired accuracy.	2nd CMs 2nd Fix n	0.
Enter value of parameter R/λ.	A	R/λ
Enter N.	R/S	$2\pi R/(\lambda N)$
To calculate I(Θ) for a given single value, enter Θ in radians.	B	I(Θ)° Allow about 50 seconds for calculation.
To calculate consecutive values of I(Θ) in steps of ΔΘ:		
Enter Θ_0.	C	Θ_0.
Enter Θ_f.	R/S	Θ_f
Enter ΔΘ and begin calculation.	R/S R/S	Θ_0° I(Θ)° Calculation of I(Θ) will take several seconds each time.
Continue to press R/S until all values of $I(\Theta_i)$ from Θ_0 to Θ_f have been calculated.	R/S R/S . . . R/S R/S	Θ_i° $I(\Theta_i)$° . . . Θ_f° $I(\Theta_f)$°

°This value is printed if the PC-100A is used.

11

Fraunhofer Diffraction at a Circular Aperture

Sample Problem

Given that:

$R/\lambda = 10$ $\quad N = 15$

$\Theta = 0$ to 0.12 radians

$\Delta\Theta = .005$ radians

find $I(\Theta_i)$ for all values of Θ.

Procedure	Press	Display/Comments
Enter program.		**0**
Set decimal to 5 places.	2nd fix 5	**0.00000**
Enter R/λ.	10 A	**10.00000**
Enter N.	15 R/S	**4.18879** $2\pi R/(N\lambda)$
Enter Θ_0.	0 C	**0.00000**
Enter Θ_f.	.12 R/S	**0.12000**
Enter $\Delta\Theta$.	.005 R/S	**0.00000** Θ_0
	R/S	**1.00000** $I(\Theta_0)$
	R/S	**0.00500** Θ_1
	R/S	**0.97555** $I(\Theta_1)$
	. . .	• • •
Continue until all values of $I(\Theta_i)$ from Θ_0 to Θ_f have been generated.	R/S	**0.12000** Θ_f
	R/S	**0.00139** $I(\Theta_f)$

The generated values are as follows:

Θ	$I(\Theta)$
0.00000	1.00000
0.00500	0.97555
0.01000	0.90517
0.01500	0.79725
0.02000	0.66423
0.02500	0.52053
0.03000	0.38034
0.03500	0.25554

Θ	$I(\Theta)$
0.04000	0.15427
0.04500	0.08023
0.05000	0.03285
0.05500	0.00818
0.06000	0.00019
0.06500	0.00223
0.07000	0.00835
0.07500	0.01415
0.08000	0.01715

Θ	$I(\Theta)$
0.08500	0.01670
0.09000	0.01351
0.09500	0.00899
0.10000	0.00463
0.10500	0.00151
0.11000	0.00010
0.11500	0.00025
0.12000	0.00139

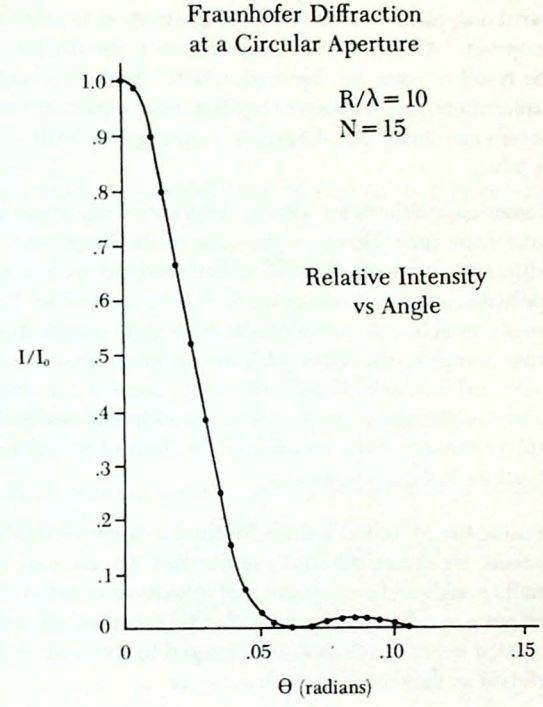
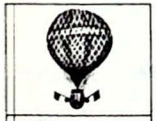


Figure 2.



FURTHER EXPLORATIONS

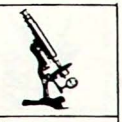
1. If the parameter R/λ is changed, how does the ratio I_1/I_0 change? I_1 is the peak intensity of the first circular fringe surrounding the central bright spot in the diffraction pattern.
2. Using a method analogous to the one given here, write a program and calculate the diffraction pattern of a rectangular aperture.

REFERENCES/ INFORMATION

See treatments of Fraunhofer diffraction in:

J. M. Stone, *Radiation and Optics: An Introduction to the Classical Theory* (New York: McGraw-Hill Book Company, Inc., 1963).

11 Euler's Method and Studies of Motion



OBJECTIVE

There are times in basic physical studies where the exploration of a particular phenomenon or physical situation reaches the point of a major "cop-out." At such times the instructor is usually forced into jumping right to the result or using an "approximation." Such "cop-outs" often occur in explorations that lead up to a differential equation since the realities of the system may make the differential equation difficult — or even impossible — to solve.

Numerical methods for solving differential equations have been around for quite some time. However, because of the large amount of iterative arithmetic involved, instructors and students seldom consider their application unless a computer is readily accessible. Here's where the programmable calculator can be extremely useful. You can use the programmable calculator with the program given here to implement numerical methods. Thus, even with complex dynamic situations you can obtain amazingly accurate results quickly and easily. In this problem we'll explore the use of the "modified" or "improved" Euler Method in several situations in basic dynamics.

FACTS/ THEORY

In using the Modified Euler's Method to analyze the motion of physical systems, we divide time into small intervals, each of length Δt. If Δt is made small enough and we assume the velocity is constant over the interval, we can get a good approximation. For this method we assume the velocity is constant over the interval and is equal to its value at the midpoint of the interval as shown in *Figure 1.*

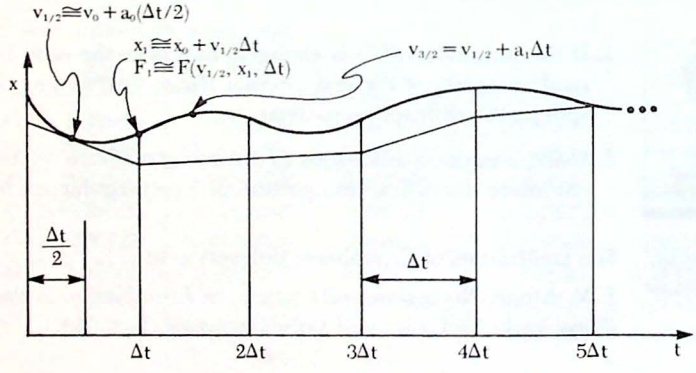


Figure 1. *Improved Euler Method*

We begin computation with the following initial conditions:

$$\begin{aligned} t &= 0 \\ x &= x_0 \\ v &= v_0 \\ F &= F_0 = F(v_0,x_0,0) \\ a_0 &= F_0/m \end{aligned}$$

Using these values, we compute the velocity value at the midpoint of the first interval as

$$v_{1/2} \cong v_0 + a_0\left(\frac{\Delta t}{2}\right)$$

Then the position at time Δt is given by

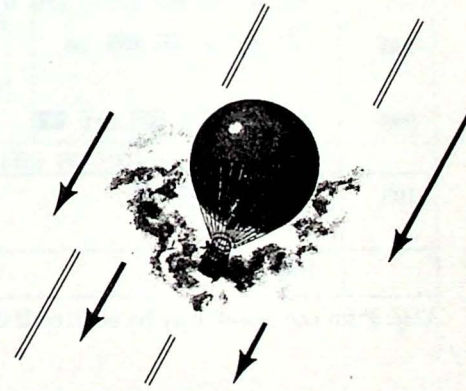
$$x_1 \cong x_0 + v_{1/2}\Delta t$$

Note that we assumed the velocity of the particle throughout the interval is constant and equal to the velocity value at time $\Delta t/2$.

Using this same procedure for every interval gives us the following general equations for $n = 1, 2, \ldots$

$$\begin{aligned} F_n &\cong F(v_{n-0.5},x_n,n\Delta t) \\ a_n &\cong \frac{F_n}{m} \\ v_{n+0.5} &\cong v_{n-0.5} + a_n\Delta t \\ x_{n+1} &\cong x_n + v_{n+0.5}\Delta t \end{aligned}$$

The following program allows you to apply the Modified Euler's Method to a wide variety of dynamic situations. To use the program you must enter the force law as subroutine C before the program is run.



11

Euler's Method and Studies of Motion

The Program

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A STO 0 R/S	Stores Δt.
005	2nd Lbl B STO 1 R/S	Stores x_0.
010	2nd Lbl C STO 2 R/S	Stores v_0.
015	2nd Lbl D STO 3 R/S	Stores m.
020	2nd Lbl E STO 4 R/S	Stores F_0.
025	2nd Lbl 2nd A' RCL 0 STO 5 CLR STO 6 STO 07	Computes $v_{1/2}$ and x_1. Initializes registers for calculations.
036	RCL 4 ÷ RCL 3 = × RCL 0 ÷ 2 = + RCL 2 = STO 07	Computes $v_{1/2} = v_0 + a_0(\frac{\Delta t}{2})$ where $a_0 = F_0/m$.
054	× RCL 0 = + RCL 1 = STO 06	Computes $x_1 = x_0 + v_{1/2}\Delta t$.
064	2nd Lbl 2nd B'	
066	RCL 5 2nd Prt R/S	Displays $n\Delta t$.
070	RCL 6 2nd Prt R/S	Displays x_n.
074	RCL 7 2nd Prt 2nd Adv R/S	Displays $v_{n-0.5}$.
079	2nd C' ÷ RCL 3 = × RCL 0 + RCL 7 = STO 07	Computes $v_{n+0.5} = a_n\Delta t + v_{n-0.5}$ where $a_n = F_n/m$.
093	× RCL 0 = SUM 06	Computes $x_{n+1} = v_{n+0.5}\,\Delta t + x_n$.
099	RCL 0 SUM 5 GTO 2nd B'	Computes $n\Delta t$. Transfers to B' to display new values.
105	2nd Lbl 2nd C'	Label for subroutine that computes the force, F_n.
	LRN RST	Exits learn mode.

Note: Print commands may be omitted if the PC-100A is not used.

REGISTER CONTENTS				
	R_{00}	Δt	R_{05}	$n\Delta t$
	R_{01}	x_0	R_{06}	x_n
	R_{02}	v_0	R_{07}	$v_{n-0.5}$
	R_{03}	m	R_{08}	
	R_{04}	F_0	R_{09}	

USING THE PROGRAM

Procedure	Press	Display/Comments
Enter subroutine to compute force. Be sure to put INV SBR as the last step.	GTO 2nd C'	0.
	LRN	107 00
	Enter keystrokes for force computation	
	INV SBR	
	LRN	0.
Enter Δt.	A	Δt
Enter x_0.	B	x_0
Enter v_0.	C	v_0
Enter m.	D	m
Enter F_0.	E	F_0
Begin calculations.	2nd A'	Δt°
	R/S	x_1°
	R/S	$v_{1/2}$°
	R/S	$2\Delta t$°
	R/S	x_2°
	R/S	$v_{3/2}$°
	. . .	
Continue to press R/S until you have calculated the desired number of values.		

°This value is printed if the PC-100A is used.

11

Euler's Method and Studies of Motion

Sample Problem

Consider a point mass, m, leaving a plane with zero initial downward velocity, as shown in *Figure 2*. The body is pulled downward by gravity and slowed in its fall by an upward viscous force proportional to its speed squared. The force acting on the body is given by

$$F = mg - kv^2$$

where k is the constant that describes the damping force, and the body's motion is determined by the differential equation

$$\frac{d^2x}{dt^2} = g - \frac{k}{m}v^2$$

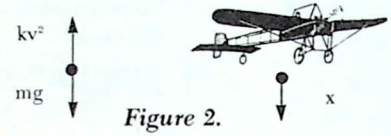


Figure 2.

We can use Euler's Method to explore the motion of the body in its downward fall. To use the program we simply need to program the force equation in subroutine C'. Then it is possible to "watch" the speed and fall distance of the body as time progresses. Use the program to watch the motion of a skydiver when the damping constant k equals 0.005 Ns^2m^{-2}. Use $\Delta t = 1s$ and $m = 1$ kg. Note that $x_0 = v_0 = 0$ and $F_0 = 9.8N$.

Procedure	Press	Display/Comments
Enter program.		0
Enter subroutine for computing force. Note that R_8 is used for storing the value of k.	GTO 2nd C'	0.
	LRN	107 00
	RCL 7 x^2	
	× RCL 8	
	= +/- +	
	RCL 3 ×	
	9.8 =	
	INV SBR	124 00
	LRN	0.
Enter Δt.	1 A	1.
Enter x_0.	0 B	0.
Enter v_0.	0 C	0.
Enter m.	1 D	1.
Enter F_0.	9.8 E	9.8
Store value of k.	.005 STO 08	0.005

Procedure	Press	Display/Comments
Set decimal place for desired accuracy.	2nd fix 3	0.005
Begin calculations.	2nd A"	1.000 Δt
	R/S	4.900 x_1
	R/S	4.900 $v_{1/2}$
	R/S	2.000 $2\Delta t$
	R/S	19.480 x_2
	R/S	14.580 $v_{3/2}$

Continue to press R/S until all desired values are computed.

The following values were obtained for n = 1, . . . , 20.

1.000	6.000	11.000	16.000
4.900	147.823	362.538	583.526
4.900	39.049	43.961	44.255
2.000	7.000	12.000	17.000
19.480	189.048	406.636	627.788
14.580	41.225	44.098	44.263
3.000	8.000	13.000	18.000
42.797	231.576	450.811	672.055
23.317	42.527	44.175	44.267
4.000	9.000	14.000	19.000
73.196	274.860	495.029	716.324
30.399	43.285	44.218	44.269
5.000	10.000	15.000	20.000
108.774	318.577	539.271	760.594
35.578	43.717	44.242	44.270

The graphs in *Figures 3* and *4* illustrate the basic physics of the system. In free fall (k = 0) the only force acting on the body is the constant force due to gravity. Thus the distance curve is parabolic in shape, and the velocity is directly proportional to time. However, if the damping force is considered, the net force is decreased as time goes on by the damping force. At the point where kv^2 equals mg, the particle reaches a terminal velocity, v_t, and the distance assumes a linear time behavior. We can solve for the terminal velocity by setting force equal to zero.

$$F = mg - kv^2$$
$$0 = mg - kv_t^2$$
$$v_t = \sqrt{\frac{mg}{k}}$$

In our case $v_t = \sqrt{\frac{9.8}{0.005}} = 44.272$ m/sec, which agrees with our results.

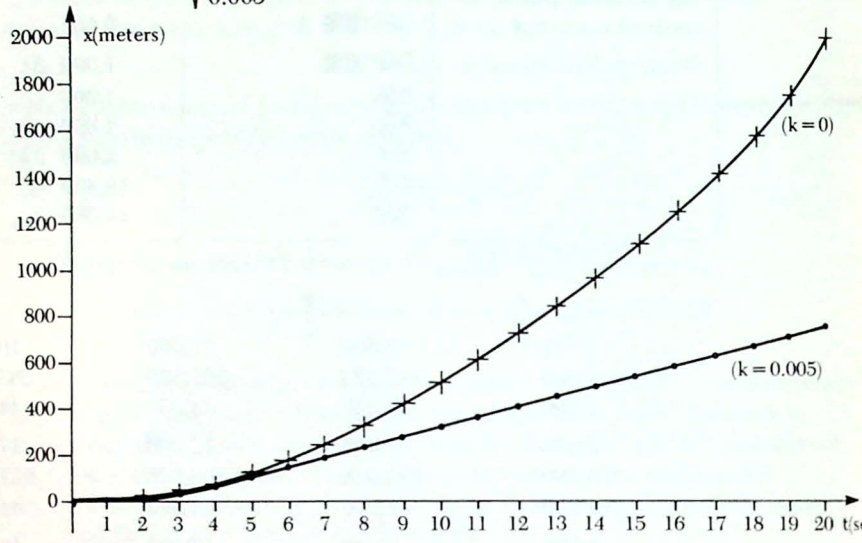


Figure 3. Distance Plot for Sample Problem

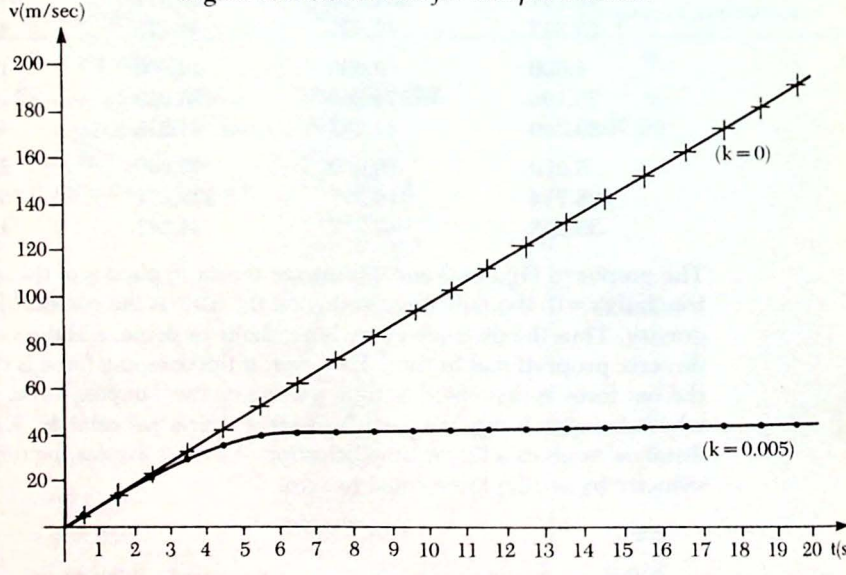


Figure 4. Velocity Plot for Sample Problem

FURTHER EXPLORATIONS

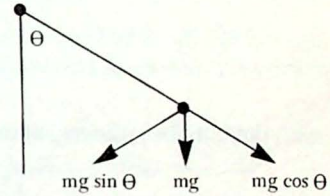
1. Using the program, plot a family of curves for a skydiver using damping constants $k = 1.0, 0.5, 0.25, 0.05, 0.01$, and $0.005\ Ns^2m^{-2}$.

   What conclusions can you draw from these graphs about the relationship between v_t and k? Could you estimate k for a steel sphere falling through water or motor oil?

2. Use the Euler's Method program to plot the same family of curves for an object falling through a medium with a linear damping force $(-kv)$; and one with a cubic damping force $(-kv^3)$. Compare the shapes of these curves to those curves for the v^2 force.

3. Use the Euler's Method program to analyze the motion of a pendulum where

$$F = -mg \sin \Theta$$



Note that the exact solution to this problem is rarely explored in basic courses. Usually, the assumption is made that Θ will remain very small, so that $\sin \Theta = \Theta$ is a valid approximation. This yields the approximate solution

$$\Theta = \Theta_{max} \cos(\omega t + \delta)$$

where

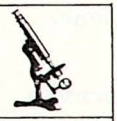
$$\omega = \sqrt{\frac{g}{l}}$$

Compare this solution to the one generated by the Euler Program. For what values of Θ is the difference significant? Note: Be sure to use radian mode for this application.

REFERENCES / INFORMATION

Robert M. Eisberg, *Applied Mathematical Physics with Pocket Programmable Calculators* (New York: McGraw-Hill Book Company, 1976), pp. 45-63.

11 Central Force Motion



OBJECTIVE

In this problem we'll be studying the dynamics of central force motion, where we use the calculator to simulate the orbit for a variety of initial conditions. For this example we'll use a more sophisticated numerical method than the Euler approach used in the previous example, called Cowell's method. The force law is entered as a separate subroutine in the program, which allows study of motional results for various force laws and constants of the motion.

FACTS/ THEORY

We wish to solve for $\vec{r}$ (t), where $\vec{r}$ satisfies

$$\frac{d^2\vec{r}}{dt^2} = \frac{1}{m}\vec{F}(\vec{r})$$

and the force $\vec{F}$ is central, i.e., parallel to the position vector $\vec{r}$. In the orbital plane we can write the following two equations:

$$\frac{d^2x}{dt^2} = f_1(x,y)$$

$$\frac{d^2y}{dt^2} = f_2(x,y)$$

with the initial conditions, at time $t = t_0$.

$$x(t_0) = x_0$$
$$y(t_0) = y_0$$
$$(v_x)_0 = \left(\frac{dx}{dt}\right)_0 = \dot{x}(t_0) = \dot{x}_0$$
$$(v_y)_0 = \left(\frac{dy}{dt}\right)_0 = \dot{y}(t_0) = \dot{y}_0$$

To solve this problem numerically, we'll use Cowell's method, which is a fourth-order predictor-corrector method for solving differential equations of this type. We will discuss the treatment of the x variable here, but keep in mind that the program will also include a similar treatment for the y dimension.

To use this method we divide the study of motion into small steps of time, each of fixed duration h. So

$$h = t_{n+1} - t_n = t_n - t_{n-1}$$

It is also assumed that we know two consecutive values of the variable, say x_{n-1} and x_n.

Following Cowell's method, we obtain a first estimate for x_{n+1} using the formula

$$\tilde{x}_{n+1} = 2x_n - x_{n-1} + h^2 f_{1,n}$$

where $f_{1,n} = f_1(x_n, y_n)$

This preliminary estimate for $\tilde{x}_{n+1}$ is then corrected (only one time in this program) by applying the "corrector"

$$x_{n+1} = 2x_n - x_{n-1} + h\, f_{1,n} + \frac{h^2}{12}(\tilde{f}_{1,n+1} - 2f_{1,n} + f_{1,n-1})$$

where $\tilde{f}_{1,n+1} = f(\tilde{x}_{n+1}, \tilde{y}_{n+1})$.

In some applications of Cowell's method, this corrector would be applied iteratively until the change in the result reached a specified tolerance. In most cases, however, one application of the corrector is sufficient.

Unfortunately, this predictor-corrector method is not self-starting. We use a method for starting given by Dahlquist and Bjork which is equivalent to solving the non-linear equations

$$x_1 = x_0 + h\,\dot{x}_0 + h^2(3f_{1,1} + 10f_{1,0} - f_{1,-1})/24$$
$$x_{-1} = x_0 - h\,\dot{x}_0 + h^2(3f_{1,-1} + 10f_{1,0} - f_{1,1})/24$$

In the program this system of equations is solved by direct iterations from the starting values

$$x_1 = x_0 + h\,\dot{x}_0 + \tfrac{1}{2}h^2 f_{1,0}$$
$$x_{-1} = x_0 - h\,\dot{x}_0 + \tfrac{1}{2}h^2 f_{1,0}$$

Once x_1 and x_{-1} are determined, the recursive step-by-step integration of the equation of motion can begin.

THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON [LRN]	Clears calculator and enters learn mode.
000	[2nd] [Lbl] [2nd] [A']	
	[RCL] 11 [2nd] [Etc] 9 [STO] 11	$t_{-1} \rightleftarrows t_1$
	[RCL] 12 [2nd] [Etc] 16 [STO] 12	$x_{-1} \rightleftarrows x_1$
	[RCL] 13 [2nd] [Etc] 17 [STO] 13	$y_{-1} \rightleftarrows y_1$
	[INV] [SBR]	

Step #	Program Keystrokes	Purpose/Comments
021	2nd Lbl A STO 10 STO 7 STO 8 R/S STO 11 STO 9 R/S STO 12 STO 14 STO 16 R/S STO 13 STO 15 STO 17 R/S 2nd Prd 7 R/S 2nd Prd 8 R/S	 Stores h. Stores t_0. Stores x_0. Stores y_0. Stores $h\dot{x}_0$. Stores $h\dot{y}_0$
055	0 x:t E RCL 18 STO 20 RCL 19 STO 21	Clears t-register. Calls subroutine E to compute $f_{1,0}$ and $f_{2,0}$.
066	RCL 10 SUM 11 INV SUM 9 x^2 STO 5 ÷ 2 = STO 6 × RCL 20 = SUM 12 SUM 16 RCL 6 × RCL 21 = SUM 13 SUM 17 RCL 7 SUM 12 INV SUM 16 RCL 8 SUM 13 INV SUM 17	Stores $t_1 = t_0 + h$ in R_{11} and $t_{-1} = t_0 - h$ in R_9. Stores h^2 in R_5. Computes $x_1 = x_0 + h\dot{x}_0 + \frac{1}{2}h^2f_{1,0}$ $x_{-1} = x_0 - h\dot{x}_0 + \frac{1}{2}h^2f_{1,0}$ $y_1 = y_0 + h\dot{y}_0 + \frac{1}{2}h^2f_{2,0}$ $y_{-1} = y_0 - h\dot{y}_0 + \frac{1}{2}h^2f_{2,0}$
113	10 STO 02 12 INV 2nd Prd 06	Loop index. Stores $h^2/24$ in R_6.
122	2nd Lbl B 2nd A' E RCL 18 STO 22 RCL 19 STO 23 2nd A' E	Subroutine to find starting values. Computes $f_{1,-1}$ and $f_{2,-1}$ and stores them in R_{22}, R_{23}. Computes $f_{1,1}$ and $f_{2,1}$.
136	RCL 18 +/- STO 16 +/- × 3 = STO 24 RCL 19 +/- STO 17 +/- × 3 = STO 25 RCL 20 × 10 = SUM 16 SUM 24 RCL 21 × 10 = SUM 17 SUM 25 RCL 22 INV SUM 24 × 3 = SUM 16 RCL 23 INV SUM 25 × 3 = SUM 17	Computes: $(3f_{1,1} + 10f_{1,0} - f_{1,-1})$ $(3f_{1,-1} + 10f_{1,0} - f_{1,1})$ $(3f_{2,1} + 10f_{2,0} - f_{2,-1})$ $(3f_{2,-1} + 10f_{2,0} - f_{2,1})$

Step #	Program Keystrokes	Purpose/Comments
198	RCL 6 2nd Prd 16 2nd Prd 17 2nd Prd 24 2nd Prd 25 RCL 7 SUM 24 INV SUM 16 RCL 8 SUM 25 INV SUM 17	Multiplies equations above by $h^2/24$ and adds or subtracts $h\dot{x}_0$ and $h\dot{y}_0$ as necessary.
222	RCL 14 SUM 16 + RCL 24 = INV SUM 12 2nd Exc 12 STO 24	Completes calculation of new x_1. Stores change in x_1 in R_{24}, then stores new x_1 in R_{12}.
237	RCL 15 SUM 17 + RCL 25 = INV SUM 13 2nd Exc 13 + RCL 24 =	Completes calculation of new y_1 and stores it in R_{13}. Computes the sum of the change in x_1 and y_1.
254	2nd x=t 260	If the sum of the changes in x_1 and y_1 equals zero, transfers out of loop.
257	2nd Dsz 2 B	Loop set for maximum of ten iterations.
260	E 2 2nd Prd 6 GTO D	Computes $f_{1,1}$ and $f_{2,1}$. Stores $h^2/12$ in R_6. Transfers to D to display results.
266	2nd Lbl C RCL 10 SUM 11 RCL 12 2nd Exc 14 2nd Exc 16 RCL 13 2nd Exc 15 2nd Exc 17 RCL 18 2nd Exc 20 2nd Exc 22 RCL 19 2nd Exc 21 2nd Exc 23	Subroutine to do integration step. Computes $t_{n+1} = t_n + h$. $x_n \to x_{n-1}$; $x_{n+1} \to x_n$ $y_n \to y_{n-1}$; $y_{n+1} \to y_n$ $f_{1,n} \to f_{1,n-1}$; $f_{1,n+1} \to f_{1,n}$ $f_{2,n} \to f_{2,n-1}$; $f_{2,n+1} \to f_{2,n}$
296	RCL 12 × 2 − RCL 16 + RCL 5 × RCL 20 = STO 12 RCL 13 × 2 − RCL 17 + RCL 5 × RCL 21 = STO 13 E	Computes x_{n+1} and y_{n+1}. Calls subroutine E to compute $f_{1,n+1}$ and $f_{2,n+1}$.

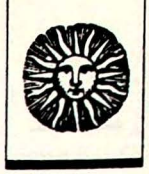
Step #	Program Keystrokes	Purpose/Comments
329	(RCL 18 − 2 × RCL 20 + RCL 22) × RCL 6 = SUM 12 (RCL 19 − 2 × RCL 21 + RCL 23) × RCL 6 = SUM 13 E	Applies corrector. Computes x_{n+1} and y_{n+1}. Calls subroutine E to compute $f_{1,n+1}$ and $f_{2,n+1}$ with corrected values.
366	2nd Lbl D RCL 11 R/S RCL 12 R/S RCL 13 R/S	Displays t_{n+1}, x_{n+1}, and y_{n+1}.
377	2nd Lbl E	Label for force law subroutine.
	LRN RST	Exits learn mode.

REGISTER CONTENTS

R_{00}		R_{09}	$t_0 - h$	R_{18}	$f_{1,n+1}$
R_{01}		R_{10}	h	R_{19}	$f_{2,n+1}$
R_{02}	Loop index	R_{11}	t_{n+1}	R_{20}	$f_{1,n}$
R_{03}		R_{12}	x_{n+1}	R_{21}	$f_{2,n}$
R_{04}		R_{13}	y_{n+1}	R_{22}	$f_{1,n-1}$
R_{05}	h^2	R_{14}	x_n	R_{23}	$f_{2,n-1}$
R_{06}	$h^2/24$; $h^2/12$	R_{15}	y_n	R_{24}	
R_{07}	$h\dot{x}_0$	R_{16}	x_{n-1}	R_{25}	
R_{08}	$h\dot{y}_0$	R_{17}	y_{n-1}	R_{26}	

USING THE PROGRAM

Procedure	Press	Display/Comments
Enter keystrokes for function subroutine. Use registers 11, 12, and 13 for t, x, and y. Store f_1 and f_2 in registers 18 and 19. Be sure and end with INV SBR	GTO E LRN keystrokes for f_1 and f_2 INV SBR LRN	379 00 0
Enter step size, h.	A	h
Enter t_0.	R/S	t_0
Enter x_0.	R/S	x_0
Enter y_0.	R/S	y_0



Procedure	Press	Display/Comments
Enter $\dot{x}_0$.	R/S	$\dot{x}_0$
Enter $\dot{y}_0$.	R/S	$\dot{y}_0$
Calculate starting values x_{-1}, x_1, y_{-1}, y_1.	R/S	t_1 t_0+h
Display new x_1	R/S	x_1
Display new y_1.	R/S	y_1
Take integration step.	C	t_2 t_0+2h
Display x_2.	R/S	x_2
Display y_2.	R/S	y_2
To obtain more points, repeat procedure for integration step as many times as necessary.	C R/S R/S	t_{n+1} $t_0+(n+1)h$ x_{n+1} y_{n+2}

SAMPLE PROBLEM

Calculate the orbits for the $1/r^2$ force law. The equations to be solved are

$$f_1 = \frac{d^2x}{dt^2} = \frac{-x}{(x^2+y^2)^{3/2}} \qquad [1]$$

$$f_2 = \frac{d^2y}{dt^2} = \frac{-y}{(x^2+y^2)^{3/2}} \qquad [2]$$

Note that time is the independent variable. Plot two orbits using the following initial velocities

inner orbit	$\dot{x}_0 = 0$
	$\dot{y}_0 = 1$
outer orbit	$\dot{x}_0 = 0$
	$\dot{y}_0 = 1.15$

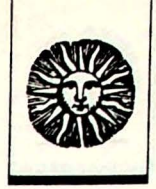
Use a step size of $\pi/20$ for both orbits and begin both trajectories at $x=1$, $y=0$.

11

Central Force Motion

Procedure	Press	Display/Comments
Enter program.		0
Enter keystrokes for f_1 and f_2.	GTO E LRN (RCL 12 x^2 + RCL 13 x^2) y^x 1.5 = 1/x STO 4 +/− × RCL 12 = STO 18 RCL 13 +/− × RCL 4 = STO 19 INV SBR LRN	379 00 413 00 0
Fix decimal for desired accuracy.	2nd fix 3	0.000
Enter step size, h.	2nd π ÷ 20 = A	 0.157
Enter t_0.	0 R/S	0.000
Enter x_0.	1 R/S	1.000
Enter y_0.	0 R/S	0.000
Enter $\dot{x}_0$.	0 R/S	0.000
Enter $\dot{y}_0$.	1 R/S	1.000
Calculate starting values.	R/S	0.157 t_1
Display x_1.	R/S	0.988 x_1
Display y_1.	R/S	0.156 y_1
Take integration step.	C	0.314 t_2
Display x_2.	R/S	0.951 x_2
Display y_2.	R/S	0.309 y_2
Take integration step.	C	0.471 t_3
Display x_3.	R/S	0.891 x_3
Display y_3.	R/S	0.454 y_3
Continue until you have the entire orbit.	• • •	• • •

This calculation takes about 95 seconds



Procedure	Press	Display/Comments
Take integration step.	C	6.126 t_{39}
Display x_{39}.	R/S	0.988 x_{39}
Display y_{39}.	R/S	−0.156 y_{39}
Take integration step.	C	6.283 t_{40}
Display x_{40}.	R/S	1.000 x_{40}
Display y_{40}.	R/S	0.000 y_{40}
For the outer orbit, clear the memories and begin by entering the step size and initial values for the outer orbit.		

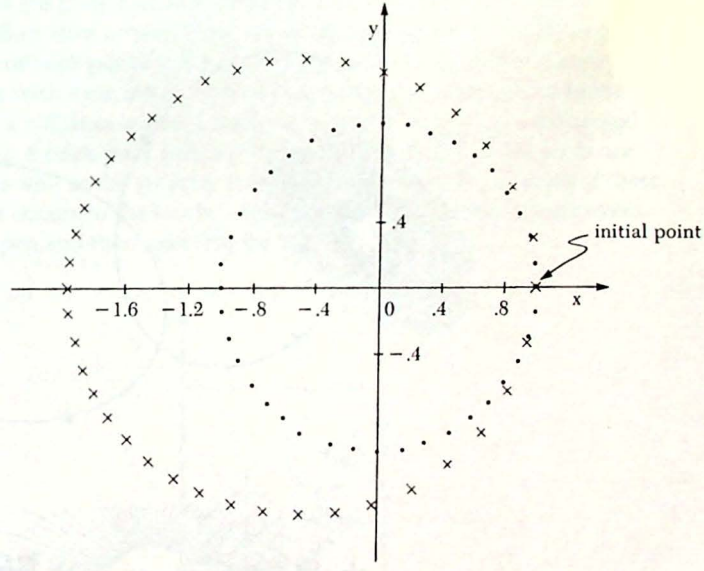


Figure 1. Inverse square force.

FURTHER EXPLORATIONS

1. What is the period of each of the orbits shown in *Figure 1*?
[*Answers:* Outer = 11.267, Inner = 6.2832 (in nondimensional time units).]

2. Let a planet be in orbit about the sun. Show how to reduce the problem to the non-dimensional form used in equations [1] and [2] of the *Sample Problem*. In terms of the parameters of the problem, what are the values of one unit of time? distance? speed?

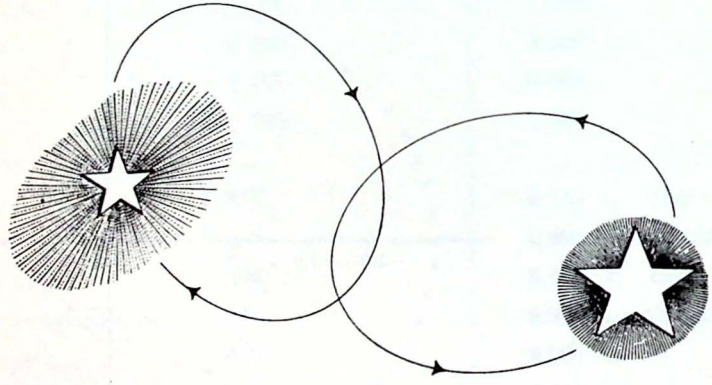
REFERENCES/ INFORMATION

Cowell's Method is described in

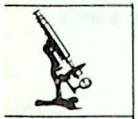
G. Dahlquist and A. Bjork, *Numerical Methods* (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1974), pp. 352-357.

For an intermediate treatment of central force motion see, for example,

K. R. Symon, *Mechanics*, 2nd ed. (Reading, Mass.: Addison-Wesley Publishing Company, Inc., 1960).



Two-Dimensional Speed Distribution



OBJECTIVE

In this project the calculator is used to numerically simulate the motion of two hard spheres bouncing within a square, two-dimensional box. This numerical experiment can be used to learn about distribution functions, to learn about the dynamics of ideal hard sphere collisions, and to get a feeling for the number of collisions which are required to establish a "good" speed distribution function in a gas. You can use the program given here to explore the dependence of the distribution function on the diameter of the spheres.

FACTS/ THEORY

We consider our "gas" to be two hard pucks each of radius a and confined in a square box of side length 1 + a as shown in *Figure 1.* The pucks shall be labelled "1" and "2" and have velocities.

$$\vec{v}_1 = v_{1,x}\vec{i} + v_{1,y}\vec{j}$$
$$\vec{v}_2 = v_{2,x}\vec{i} + v_{2,y}\vec{j} \qquad [1$$

Each puck's trajectory is described by two equations

$$x_i(t) = x_{i0} + v_{i,x}t \qquad i = 1, 2, \ldots$$
$$y_i(t) = y_{i0} + v_{i,y}t \qquad [2$$

where t is the time.

If the pucks are launched with arbitrary initial velocities at arbitrary positions, than after a short time one of the following events will happen. Either one or both pucks will collide with the walls of the box, or the pucks will collide with each other. Any of these collisions is assumed to be elastic. In the case of a collision with the wall, the speed of the puck is not changed. However, in a puck-puck collision the individual speeds of the pucks are changed, as well as the velocity components. We determine which of these events next occurs in the box by calculating the time at which each event would happen and then selecting the earliest one.

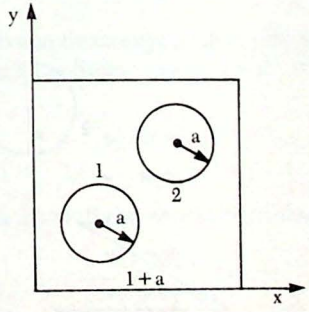


Figure 1. *Two-Dimensional "Gas" of Two Frictionless Pucks.*

The time $t_{x,i}$ of a collision with a wall which is perpendicular to the x axis (which will be called an x-wall) is given by

$$t_{x,i} = \frac{1}{v_{i,x}} \left[\begin{Bmatrix} 0, & v_{i,x} < 0 \\ 1, & v_{i,x} > 0 \end{Bmatrix} - x_{i0} \right] \qquad [3]$$

and similarly the time $t_{y,i}$ for a collision with a y-wall is

$$t_{y,i} = \frac{1}{v_{i,y}} \left[\begin{Bmatrix} 0, & v_{i,y} < 0 \\ 1, & v_{i,y} > 0 \end{Bmatrix} - y_{i0} \right] \qquad [4]$$

The time t_c for a puck-puck collision is found from the condition that the distance between the centers of the two pucks be exactly equal to the puck diameter, or

$$(y_2 - y_1)^2 + (x_2 - x_1)^2 = 4a^2 \qquad [5]$$

Substituting from equations [2] we arrive at a quadratic equation

$$At_c^2 + Bt_c + C = 0 \qquad [6]$$

where

$$\begin{aligned} A &= (v_{2,y} - v_{1,y})^2 + (v_{2,x} - v_{1,x})^2 \\ B &= 2\left[(y_{20} - y_{10})(v_{2,y} - v_{1,y}) + (x_{20} - x_{10})(v_{2,x} - v_{1,x})\right] \\ C &= (y_{20} - y_{10})^2 + (x_{20} - x_{10})^2 - 4a^2 \end{aligned} \qquad [7]$$

The solutions are

$$t_c = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \qquad [8]$$

These solutions for the collision time may be real or complex depending on the algebraic sign of the expression $(B^2 - 4AC)$. If the calculated values of t_c are complex, then in the program it is assumed that some other event will occur next. If there are real solutions, we will always want the smallest one since this is the one which occurs when the pucks first touch. The larger real solution is unrealizable as a future event, though it describes a collision in the past.

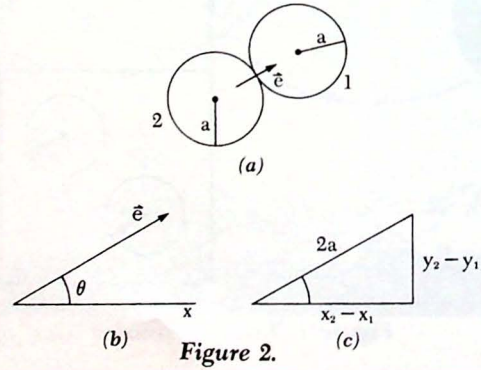


Figure 2.

In the event of a puck-puck collision the new velocities are determined by momentum and energy conservation principles:

$$\vec{v}_1 + \vec{v}_2 = \vec{v}_1' + \vec{v}_2'$$
$$v_1^2 + v_2^2 = v_1'^2 + v_2'^2 \qquad [9]$$

where the prime superscript denotes a post-collision quantity. In the case of smooth, hard spheres the solution is given by

$$\vec{v}_2' = \vec{v}_2 + (\vec{V}\cdot\hat{e})\hat{e}$$
$$\vec{v}_1' = \vec{v}_1 - (\vec{V}\cdot\hat{e})\hat{e} \qquad [10]$$

where $\vec{V}$ is the initial (pre-collision) relative velocity,

$$\vec{V} = \vec{v}_1 - \vec{v}_2 \qquad [11]$$

and $\hat{e}$ is a unit vector along the line of centers. (See *Figure 2a.*) $\hat{e}$ is inclined at an angle θ from the x axis and this angle is specified by (see *Figures 2b* and *2c*)

$$\cos\theta = \frac{x_1 - x_2}{2a}$$
$$\sin\theta = \frac{y_1 - y_2}{2a} \qquad [12]$$

Since $\hat{e}$ is a unit vector, $\vec{V}\cdot\hat{e}$ is given by

$$\vec{V}\cdot\hat{e} = (v_{1,x} - v_{2,x})\cos\theta + (v_{1,y} - v_{2,y})\sin\theta \qquad [13]$$

and the new puck velocities are

$$v_{2,x}' = v_{2,x} + (\vec{V}\cdot\hat{e})\cos\theta$$
$$v_{2,y}' = v_{2,y} + (\vec{V}\cdot\hat{e})\sin\theta$$
$$v_{1,x}' = v_{1,x} - (\vec{V}\cdot\hat{e})\cos\theta$$
$$v_{1,y}' = v_{1,y} - (\vec{V}\cdot\hat{e})\sin\theta \qquad [14]$$

In the event of a wall collision the component of velocity perpendicular to the wall is reversed. If puck i collides with an x-wall, then the new velocities are

$$v_{i,x}' = -v_{i,x}$$
$$v_{i,y}' = v_{i,y} \qquad [15]$$

and if the collision is with a y-wall the new velocities are

$$v_{i,x}' = v_{i,x}$$
$$v_{i,y}' = -v_{i,y} \qquad [16]$$

Equations [14], [15], and [16] give the new velocities after any of the possible events. We must also calculate the new constants for the trajectories. After any collision the new trajectory is specified by

$$x_i' = v_{i,x}'t + x_{i0}'$$
$$y_i' = v_{i,y}'t + y_{i0}' \qquad [17]$$

Since $x_i = x_i'$ and $y_i = y_i'$ at the instant of collision t_0, the new constants are

$$x_{i0}' = x_{i0} + (v_{i,x} - v_{i,x}')t_0$$
$$y_{i0}' = y_{i0} + (v_{i,y} - v_{i,y}')t_0 \qquad [18]$$

Continuing from collision to collision, we now have a complete, deterministic description of the motions of the two pucks. After every puck-puck collision we update the cumulative speed distribution by adding the length of time during which each puck has had its speed to the accumulant in the appropriate speed interval bin. In this way we numerically tabulate the time-averaged speed distribution for the two pucks.

For the idealized case of two-point pucks ($a = 0$) the theoretical speed distribution function $f(u)$ in two dimensions is

$$f(u)\,du = Audu \qquad 0 \le u \le u_{max}$$
$$f(u)\,du = 0 \qquad y > u_{max} \qquad [19]$$

where u is the speed, A is a normalization constant, and u_{max} is the maximum allowable speed, given by

$$u_{max}^2 = u_{10}^2 + u_{20}^2 \qquad [20]$$

where u_{10} and u_{20} are the initial speeds of the two pucks. The speed distribution function given by equation [19] is based on the assumption that all points in the allowable phase space have equal probabilities of occurrence. This assumption will not be satisfied if special, rational initial conditions are selected. For example, one such possibility has both pucks symmetrically bouncing along a common straight line between puck-puck and wall collisions. Initial conditions leading to such periodic solutions should be avoided.

THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON 4 [2nd] [Op] 17 [CLR] [LRN]	Clears calculator, repartitions the memory, and enters learn mode.
000	[GTO] 272	Resumes execution after [RST].

Step #	Program Keystrokes	Purpose/Comments
003	RCL 19 STO 2nd Ind 8 RCL 2nd Ind 6 x:t 0 2nd x=t 029 2nd x=t 018 1	Subroutine to calculate wall collision time. If v = 0, sets t = ∞. Otherwise
018	(CE − RCL 2nd Ind 7) ÷ x:t = STO 2nd Ind 8 INV SBR	$t = \left[\begin{cases} 0, v<0 \\ 1, v>0 \end{cases} - x_0 \right] \frac{1}{v}$
030	RCL 14 x^2 + RCL 16 x^2 = STO 1 $\sqrt{x}$ STO 20 RCL 15 x^2 + RCL 17 x^2 = SUM 1 $\sqrt{x}$ STO 21 INV SBR	Subroutine to calculate $\lvert\vec{v}_1\rvert$ and $\lvert\vec{v}_2\rvert$ $v_i = \left[v_{i,x}^2 + v_{i,y}^2 \right]^{1/2}$
057	2nd Lbl A 2nd St flg 0 2nd St flg 1 2nd St flg 2 2nd St flg 3 SBR 030 RCL 1 $\sqrt{x}$ ÷ 10 = STO 02 1 EE 90 STO 19 0 STO 1 INV EE	Initializes routine. Sets flags to indicate all times need to be calculated. Calls subroutine 030 to compute v_1 and v_2. Sets Δv for ten bir counting. Stores 10^{90} as constant for "infinity." Sets t = 0.
090	2nd Lbl B 14 STO 06 10 STO 07 23 STO 08	Begins routine for each collision. Sets addresses for subroutine 003.
104	INV 2nd If flg 0 112 SBR 003	Calculates t_{x1} (SBR 003) if needed.
112	2nd Op 26 2nd Op 27 2nd Op 28	Increments addresses.
118	INV 2nd If flg 1 126 SBR 003	Calculates t_{x2} (SBR 003) if needed.
126	2nd Op 26 2nd Op 27 2nd Op 28	Increments addresses.
132	INV 2nd If flg 2 140 SBR 003	Calculates t_{y1} (SBR 003) if needed.
140	2nd Op 26 2nd Op 27 2nd Op 28	Increments addresses.
146	INV 2nd If flg 3 154 SBR 003	Calculates t_{y2} (SBR 003) if needed.

Two-Dimensional Speed Distribution

Step #	Program Keystrokes	Purpose/Comments
154	RCL 19 STO 27 2nd If flg 8 271	Begins calculation of t_c. If last event was a collision (flag 8 set), goes to 271.
162	(RCL 15 − RCL 14) STO 28 x^2 + (RCL 17 − RCL 16) STO 29 x^2 = STO 03	Calculates A. (equation [7])
186	(RCL 13 − RCL 12) 2nd Prd 29 x^2 + (RCL 11 − RCL 10) 2nd Prd 28 x^2 − RCL 22 x^2 = STO 05	Calculates C. (equation [7])
214	RCL 28 + RCL 29 = × 2 = STO 04	Calculates B. (equation [7])
225	2 2nd Prd 03 0 $x{:}t$ RCL 4 x^2 − 2 × RCL 3 × RCL 5 = INV 2nd $x{=}t$ 271	If $B^2-4AC<0$, $t_c=\infty$ (no real roots).
246	$\sqrt{x}$ STO 5 + RCL 4 = +/− ÷ RCL 3 = STO 28 RCL 1 $x{:}t$ RCL 28 INV 2nd $x{=}t$ 271 STO 27	Calculates t_c. If $t_c<$ time of last event, sets $t_c=\infty$. (See step 154.) Otherwise stores t_c in R_{27}.
271	RST	Unsets all flags, then returns to next step.
272	RCL 23 $x{:}t$ RCL 24 2nd $x{=}t$ 283 2nd St flg 5 $x{:}t$	Finds the earliest event by setting a flag each time one event has an earlier time.
283	RCL 25 2nd $x{=}t$ 291 2nd St flg 6 $x{:}t$	
291	RCL 26 2nd $x{=}t$ 299 2nd St flg 7 $x{:}t$	
299	RCL 27 2nd $x{=}t$ 307 2nd St flg 8 $x{:}t$	
307	RCL 14 STO 6 RCL 15 STO 7 RCL 16 STO 8 RCL 17 STO 28 $x{:}t$ STO 01	Saves current values for velocities and time.

Step #	Program Keystrokes	Purpose/Comments
326	1 [+/−] [2nd] [If flg] 8 408 [2nd] [If flg] 7 392 [2nd] [If flg] 6 376 [2nd] [If flg] 5 360	Places −1 in display. Checks flags to find the earliest event and goes to appropriate section.
344	[2nd] [St flg] 0 [2nd] [Prd] 14 2 [×] [RCL] 6 [×] [RCL] 23 [=] [SUM] 10 [GTO] [B]	Puck 1 hits x-wall. Multiplies $v_{1,x}$ by −1. Calculates x'_{10}. Transfers to [B] to start new calculation.
360	[2nd] [St flg] 1 [2nd] [Prd] 15 2 [×] [RCL] 7 [×] [RCL] 24 [=] [SUM] 11 [GTO] [B]	Puck 2 hits x-wall. Multiplies v_{2x} by −1. Calculates x'_{20}. Transfers to [B] to start new calculation.
376	[2nd] [St flg] 2 [2nd] [Prd] 16 2 [×] [RCL] 8 [×] [RCL] 25 [=] [SUM] 12 [GTO] [B]	Puck 1 hits y-wall. Multiplies v_{1y} by −1. Calculates y'_{10}. Transfers to [B] to start new calculation.
392	[2nd] [St flg] 3 [2nd] [Prd] 17 2 [×] [RCL] 28 [×] [RCL] 26 [=] [SUM] 13 [GTO] [B]	Puck 2 hits y-wall. Multiplies v_{2y} by −1. Calculates y'_{20}. Transfers to [B] to begin new calculation.
408	[RCL] 14 [−] [RCL] 15 [=] [STO] 29 [×] [RCL] 27 [+] [RCL] 10 [−] [RCL] 11 [=] [÷] [RCL] 22 [=] [STO] 4 [INV] [2nd] [cos] [2nd] [sin] [2nd] [|x|] [STO] 05	Pucks collide. Calculates $v_{1x} - v_{2x}$. Calculates cos θ and stores it in R_4. Then computes |sin θ| and stores it in R_5.
438	[RCL] 16 [−] [RCL] 17 [=] [STO] 1 [×] [RCL] 27 [+] [RCL] 12 [−] [RCL] 13 [=] [2nd] [Op] 10 [2nd] [Prd] 5 [RCL] 05	Calculates $v_{1y} - v_{2y}$. Computes $(v_{1y} - v_{2y})t_c + (y_1 - y_2)$. Computes signum function (*Master Library*, p. V−28) and multiplies by |sin θ| to give sin θ the proper sign.

Step #	Program Keystrokes	Purpose/Comments
462	× RCL 1 + RCL 4 × RCL 29 = STO 03	Computes $\vec{V} \cdot \hat{e}_1$.
474	× RCL 4 = SUM 15 INV SUM 14	Computes $v_{2'x}$, $v_{1'x}$.
483	RCL 3 × RCL 5 = SUM 17 INV SUM 16	Computes $v_{2'y}$, $v_{1'y}$.
494	2nd St flg 0 2nd St flg 1 2nd St flg 2 2nd St flg 3	Sets flag to indicate all time must be recalculated.
502	RCL 6 − RCL 14 = × RCL 27 = SUM 10 RCL 7 − RCL 15 = × RCL 27 = SUM 11	Computes $x_{1'0}$ and $x_{2'0}$.
526	RCL 8 − RCL 16 = × RCL 27 = SUM 12 RCL 28 − RCL 17 = × RCL 27 = SUM 13	Computes $y_{1'0}$ and $y_{2'0}$.
550	20 STO 28 SBR 577 21 STO 28 SBR 577 SBR 030 RCL 27 STO 18 STO 1 2nd Dsz 9 B INV SBR	Stores address of v_1. Calls subroutine 577 to update distribution with v_1. Stores address of v_2. Calls subroutine 577 to update distribution with v_2. Calculates new v_1 and v_2. Stores the time. Transfers to B if new calculation necessary. Otherwise stops.
577	0 STO 29 10 STO 0 RCL 2nd Ind 28 x≥t	Starts with lowest speed bin. Sets loop counter. Recalls velocity value to be added to the distribution.
587	RCL 2 SUM 29 RCL 29 2nd x=t 600 2nd Dsz 0 587	$v_{bin} = v_{bin} + \Delta v$ (top of bin). Exits loop if $v \leq v_{bin}$. Tries next bin.
600	40 − RCL 0 = STO 0 RCL 27 − RCL 18 = SUM 2nd Ind 0 INV SBR	Register for this bin. Adds time elapsed since last collision to this bin.

Step #	Program Keystrokes	Purpose/Comments
617	2nd Lbl C 30 STO 00	Subroutine to recall and display the values for the speed distribution stored in $R_{30} - R_{39}$.
623	RCL 2nd Ind 0 2nd Prt R/S 2nd Op 20 GTO 623	
	LRN RST	Exits learn mode.

Note: Print command may be omitted if PC-100A is not used.

REGISTER CONTENTS

R_{00}	Loop index	R_{14}	v_{1x}	R_{28}	Used
R_{01}	$t_c, v_1^2 + v_2^2, v_{1y} - v_{2y}$	R_{15}	v_{2x}	R_{29}	Used
R_{02}	Δv	R_{16}	v_{1y}	$R_{30} - R_{39}$	Storage bins for speed distribution
R_{03}	A, $\vec{V} \cdot \vec{e}$	R_{17}	v_{2y}		
R_{04}	B, x_1, $\cos\theta$	R_{18}	t		
R_{05}	C, $\sqrt{B^2 - 4AC}$, $\sin\theta$	R_{19}	Used		
R_{06}	Used	R_{20}	v_1		
R_{07}	Used	R_{21}	v_2		
R_{08}	Used	R_{22}	2a		
R_{09}	N	R_{23}	t_{x1}		
R_{10}	x_{10}	R_{24}	t_{x2}		
R_{11}	x_{20}	R_{25}	t_{y1}		
R_{12}	y_{10}	R_{26}	t_{y2}		
R_{13}	y_{20}	R_{27}	t_c		

USING THE PROGRAM

Procedure	Press	Display/Comments
Repartition the calculator memory.	4 2nd Op 17	639.39
Enter program carefully. Be sure to check that you entered all keystrokes correctly.		
Fix decimal to desired accuracy.	2nd Fix n	
Enter the number of collisions, N.	N STO 09	N

Two-Dimensional Speed Distribution

Procedure	Press	Display/Comments
Store initial data.	x_{10} STO 10	x_{10}
	x_{20} STO 11	x_{20}
	y_{10} STO 12	y_{10}
	y_{20} STO 13	y_{20}
	v_{1x} STO 14	v_{1x}
	v_{2x} STO 15	v_{2x}
	v_{1y} STO 16	v_{1y}
	v_{2y} STO 17	v_{2y}
Store puck diameter, 2a.	STO 22	2a
Begin program execution.	A	t elapsed time
Display speed distribution values.		
Bin 1.	C	t_{1v}° time units in bin 1.
Bin 2.	R/S	t_{2v}° time units in bin 2.
Bin 3.	R/S	t_{3v}° time units in bin 3.
Bin 4.	R/S	t_{4v}° time units in bin 4.
Bin 5.	R/S	t_{5v}° time units in bin 5.
Bin 6.	R/S	t_{6v}° time units in bin 6.
Bin 7.	R/S	t_{7v}° time units in bin 7.
Bin 8.	R/S	t_{8v}° time units in bin 8.
Bin 9.	R/S	t_{9v}° time units in bin 9.
Bin 10.	R/S	t_{10v}° time units in bin 10.
If additional calculations are desired for more collisions, continue with the following. Be sure you do not alter any flags or memories.		
Enter number of additional collisions, N.	STO 09	N.
Begin calculations.	B	t total time elapsed.
Display speed distribution values if desired. (See above.)		

°This value is printed if the PC-100A is used.

SAMPLE PROBLEM

To test your entered program, run the following set of initial conditions:

Number of collisions = 5
Puck diameter, 2a = .3

$x_{10} = 0.011$ $\quad v_{1x} = 1.013$
$x_{20} = 0.902$ $\quad v_{2x} = -1.211$
$y_{10} = 0.501$ $\quad v_{1y} = 1.103$
$y_{20} = 0.302$ $\quad v_{2y} = -0.225$

Procedure	Press	Display/Comments
Enter program.		0
Fix decimal for 3 places.	[2nd] [Fix] 3	0.000
Enter number of collisions.	5 [STO] 09	5.000
Enter initial data.		
x_{10}	.011 [STO] 10	0.011
x_{20}	.902 [STO] 11	0.902
y_{10}	.501 [STO] 12	0.501
y_{20}	.302 [STO] 13	0.302
v_{1x}	1.013 [STO] 14	1.013
v_{2x}	1.211 [+/−] [STO] 15	−1.211
v_{1y}	1.103 [STO] 16	1.103
v_{2y}	.225 [+/−] [STO] 17	−0.225
Enter puck diameter.	.3 [STO] 22	0.300
Begin program execution.	[A]	4.254 elapsed time This calculation takes about ten minutes.
Recall speed distribution values.	[C]	0.348 t_{1v}
	[R/S]	0.000 t_{2v}
	[R/S]	0.000 t_{3v}
	[R/S]	0.773 t_{4v}
	[R/S]	0.972 t_{5v}
	[R/S]	0.976 t_{6v}
	[R/S]	1.185 t_{7v}
	[R/S]	1.185 t_{8v}
	[R/S]	0.976 t_{9v}
	[R/S]	2.093 t_{10v}

11

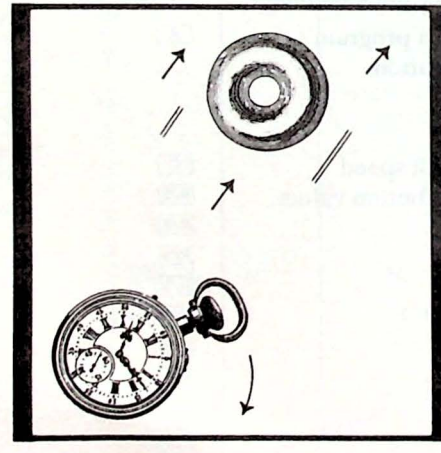
Two-Dimensional Speed Distribution

Further Explorations

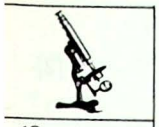
1. Suppose the program calculates N collisions in time interval t. What is the value of the constant A in Equation 14? (Don't forget that there are two pucks.)
2. If one puck has a diameter 2a = .3 and the other puck has a diameter 2a = .2, what is the value of 2a which should be entered into the program? (*Answer:* ½(.2 + .3) = .25).
3. Have the calculator tabulate distribution functions for runs with several numbers of collisions, e.g., 50, 100, 200, . . . After each run find the rms deviation of the tabulated distribution from the theoretical one. Plot this rms deviation as a function of number of collisions and discuss your results.
4. What do you expect the effect of a non-zero puck size is on the shape of the speed distribution function? Investigate this effect by making several long runs with different puck diameters.

References/ Information

A. Sommerfield, *Thermodynamics and Statistical Mechanics* (New York: Academic Press, Inc., 1964), p. 297.



The Equation of Time



OBJECTIVE

The previous sections in this chapter have suggested new ways to explore the mathematics and mathematical models of physics with the programmable calculator. The calculator can also be a valuable aid in the study of basic astronomy, offering quick computation for graphing and iterative numerical solutions. This section and the following section illustrate only a few of the wide range of applications for the programmable calculator in basic astronomy explorations.

With the help of your calculator, the Equation of Time can easily be calculated to an accuracy of a few seconds. You can directly apply and calculate the two physical effects known to be responsible, i.e., the obliquity of the ecliptic and the eccentricity of the earth's orbit.

FACTS/ THEORY

The equation of the sun's orbit about the earth is

$$\frac{1}{r} = A(1 + e \cos [\Theta - \Theta_0]) \qquad [1]$$

where e is the eccentricity, A is a constant, and Θ_0 is the sun's longitude at perihelion (see *Figure 1*).

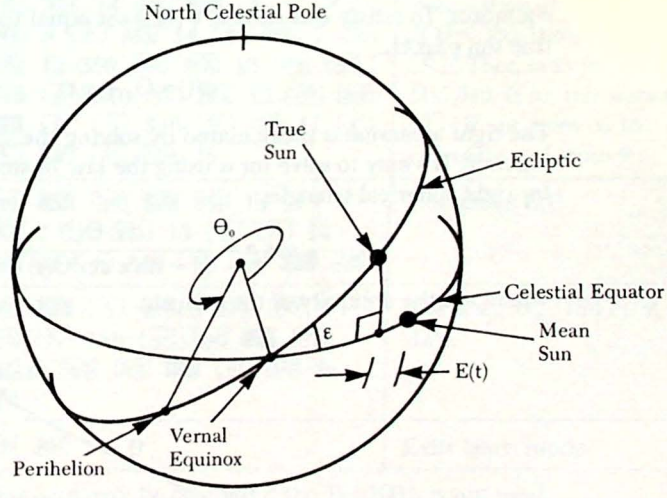


Figure 1.

The oribital angular momentum L is constant, so we have

$$\frac{d\Theta}{dt} = \frac{L}{mr^2} = \frac{LA^2}{m}\,[1 + e\cos(\Theta - \Theta_0)]^2 \qquad [2]$$

Neglecting the small e^2 term, then,

$$\frac{d\Theta}{dt} \cong \omega_0(1 + 2e\cos(\Theta - \Theta_0)) \qquad [3]$$

where $\omega_0 = LA^2/m$. ω_0 represents the sun's *average* angular velocity on the ecliptic. To solve this equation with adequate accuracy, we put $\Theta(t) = \Theta_0 + \omega_0 t$ into the right-hand side and integrate with respect to mean solar time t. Then we have

$$\Theta(t) \cong \Theta_0 + \omega_0 t + 2e\sin\omega_0 t \qquad [4]$$

The time t is measured from the instant of perihelion passage.

To find the Equation of Time, E(t), we find the sun's true progress [$\phi(t)$] along the celestial equator and compare it with the progress of the mean sun ($\omega_0 t$).

$$E(t) = \omega_0 t - \phi(t) \qquad [5]$$

The instants of time at which $E = 0$ are determined by requiring that if the eccentricity (e) is equal to zero, then E is also equal to zero at the equinoxes. To satisfy this choice, $\phi(t)$ is set equal to the right ascension of the true sun plus Θ_0.

$$\phi(t) = \Theta_0 + \alpha(t) \qquad [6]$$

The right ascension α is calculated by solving the spherical triangle shown in *Figure* 2. It's easy to solve for α using the law of sines and the law of cosines for right spherical triangles:

$$\cos\alpha = \frac{\cos\Theta(t)}{[1 - \sin^2\varepsilon\,\sin^2\Theta(t)]^{1/2}} \qquad [7]$$

where ε is the obliquity of the ecliptic.

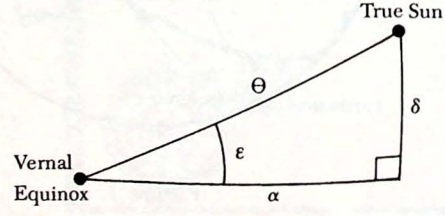


Figure 2.

The following program allows you to evaluate the Equation of Time for any year, given the astronomical constants ω_0, Θ_0, e, and ε for that epoch. Before running the program, be sure that the calculator is in the degree mode.

THE PROGRAM

Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl E RCL 11 2nd sin x^2 × RCL 9 +/− + 1 = $\sqrt{x}$ $1/x$ × RCL 11 2nd cos = INV 2nd cos STO 12 INV SBR	Calculates α (right ascension).
025	2nd Lbl C − 180 = STO 11 E + 180 = STO 15 GTO 2nd A'	Calculates ϕ (sun's true progress along celestial equator) if $\Theta < 360$.
044	2nd Lbl D − 360 = STO 11 E + 360 = STO 15 GTO 2nd A'	Calculates ϕ if $\Theta < 540$.
063	2nd Lbl B STO 5 × RCL 6 = STO 14 2nd sin × 2 × RCL 8 + RCL 14 + RCL 7 = STO 13 360 $x \rightleftarrows t$ RCL 13 INV 2nd $x \geq t$ C 540 $x \rightleftarrows t$ RCL 13 INV 2nd $x \geq t$ D − 540 = STO 11 E + 540 = STO 15	Stores t and begins calculation of ϕ. Tests to see if $\Theta < 360$; if so, transfers to C. Then tests to see if $\Theta < 540$. If so, transfers to D; if not, goes on to calculate and store ϕ.
121	2nd Lbl 2nd A' RCL 14 + RCL 7 − RCL 15 = × 24 × 60 ÷ 360 = 2nd Prt R/S	Computes E(t).
145	2nd Lbl A STO 6 R/S STO 7 R/S × 180 ÷ 2nd π = STO 8 R/S 2nd sin x^2 STO 9 R/S	Stores ω_0, Θ_0, $(180\ e)/\pi$, and $\sin^2\varepsilon$.
	LRN RST	Exits learn mode.

Note: Print command may be omitted if the PC-100A is not used.

11

The Equation of Time

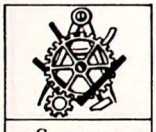
Register Contents

R_{00}		R_{06}	ω_0	R_{12}	α
R_{01}		R_{07}	Θ_0	R_{13}	Θ
R_{02}		R_{08}	$(180\ e)/\pi$	R_{14}	$\omega_0 t$
R_{03}		R_{09}	$\sin^2\varepsilon$	R_{15}	ϕ
R_{04}		R_{10}		R_{16}	
R_{05}	t	R_{11}	Θ'	R_{17}	

Using the Program

Procedure	Press	Display/Comments
Enter data (constants for epoch).		
ω_0	A	ω_0
Θ_0	R/S	Θ_0
e	R/S	$(180\ e)/\pi$
ε	R/S	$\sin^2\varepsilon$
Enter t (measured from the instant of perihelion passage) and calculate E(t).	B	E(t)° Allow several seconds for calculation.
Repeat this step as often as desired.		

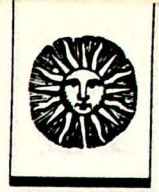
°This value is printed if the PC-100A is used.



Sample Problem

The astronomical constants for the epoch 1980 are given below. Use this information to solve the Equation of Time from t = 0 to t = 360, at intervals of 20 days.

ω_0 = 360°/365.2421982 days
Θ_0 = 282.59646°
e = 0.0167175°
ε = 23.441883°

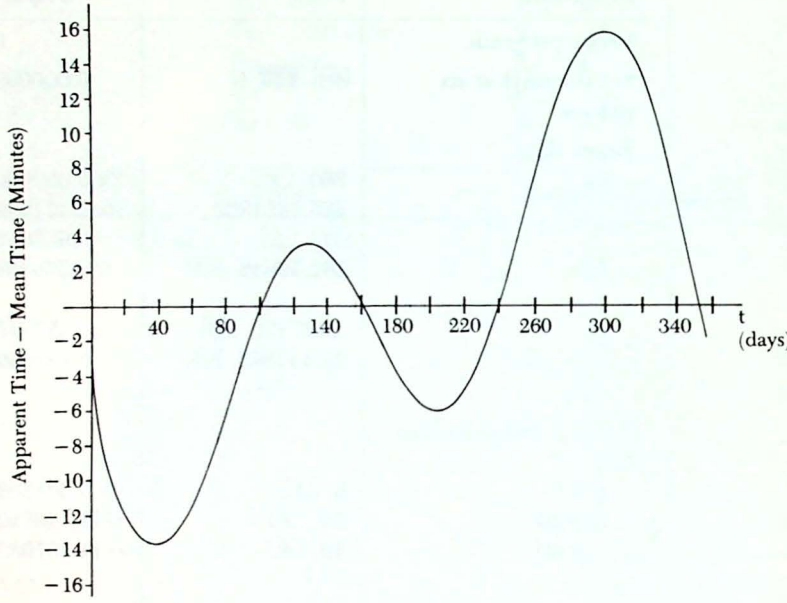


Procedure	Press	Display/Comments
Enter program.		0
Set decimal at six places.	2nd f 6	0.000000
Enter data.		
ω_0	360 ÷	360.000000
	365.2421982	365.2421982 ω_0
	= A	0.985647
Θ_0	282.59646 R/S	282.596460 Θ_0
e	.0167175 R/S	0.957842 $(180\ e)/\pi$
ε	23.441883 R/S	0.158260 $\sin^2\varepsilon$
Enter t and calculate E(t).		
$t_0 = 0$	0 B	−4.368882
$t_1 = 20$	20 B	−11.746792
$t_2 = 40$	40 B	−14.207033
	• • •	• • •
Continue for all values of t.		
$t_{18} = 360$	360 B	−1.889634

The calculated values are given in the chart and graph below:

t (*days*)	*E(t)* (*minutes*)	*t* (*days*)	*E(t)* (*minutes*)	*t* (*days*)	*E(t)* (*minutes*)
0	−4.368882	120	3.039227	260	6.456142
20	−11.746792	140	3.272447	280	12.854483
40	−14.207033	160	0.262027	300	16.223871
60	−11.796538	180	−3.894666	320	14.591118
80	−6.421670	200	−6.328610	340	7.750332
100	−0.675805	220	−5.151356	360	−1.889634
		240	−0.360710		

The Equation of Time



Further Explorations

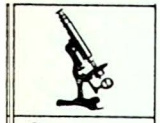
1. Derive Equation [7] by solving the right spherical triangle of *Figure* 2.
2. Investigate the effect of the values of Θ_0, e, and ε on the curve of E(t) vs. t.

References/ Information

A clear but qualitative discussion of the Equation of Time is found in the following text:

Stanley P. Wyatt, *Principles of Astronomy,* 3rd ed. (Boston: Allyn and Bacon, Inc., 1977), pp. 83-84.

Sundial Design



OBJECTIVE

The construction of a simple sundial makes an interesting special project for the general astronomy or practical astronomy course and helps to reinforce the basic facts about the positional relationship of the sun and earth.

The handheld programmable calculator speeds up the computational aspects of the design and enables you to check out many possible configurations quickly, so that the best one can be selected. By entering the parameters describing your latitude and dial face orientation, you can then compute the angle between the edge of the gnomon shadow and the line in the dial plane which lies in the north-south vertical plane.

FACTS/ THEORY

The constructional aspects of sundial design will not be treated here. (For a practical approach to this subject, see *Sundials: Their Theory and Construction* by Albert E. Waugh.) We'll confine ourselves here to the calculation of the angles of the hour lines for a sundial whose indicator or gnomon edge is parallel to the rotation axis of the earth and which has a flat, though not necessarily level, dial face. With the program given here we can find the hour lines for sundials on vertical walls, sloping ground, or other surfaces, in addition to the ordinary horizontal sundial.

If you want the sundial to indicate standard time rather than local solar time, you must incorporate a longitude correction. This is accomplished by applying a constant time shift based on the difference between the longitude at which standard time is set and the longitude of your location. Because this shift depends on location, we won't consider this factor in our discussion, but it should be kept in mind. Also, sundial readings must be corrected for the Equation of Time (see the previous section) if more than 15-minute accuracy over the entire year is desired. Unfortunately, this correction is time-dependent and is difficult to incorporate into the design. A graph is often provided with the sundial to illustrate this compensation.

For sundials in the Northern Hemisphere, the dial face orientation is specified in the following way. The dial, starting out level at latitude ϕ, is tipped along an east-west axis by angle i. (Angle i is positive if the south edge goes down; otherwise it's negative.) The face is then rotated about a vertical axis by angle ψ, where ψ is positive if the face rotates clockwise as seen from above. Thus, for example, a southeast-facing wall would be described by $i = 90°$, $\psi = -45°$, and a horizontal dial by $i = 0$, $\psi = 0$. The geometry of the situation is shown in *Figure 1.* On this diagram of the celestial sphere, P is the north celestial pole, Z is the observer's zenith, W is the west point, S is the south point, AWRS is along the horizon, WQ is the celestial equator, and AUVS is the intersection of the plane of the dial face with the celestial sphere. The hour angle is H.

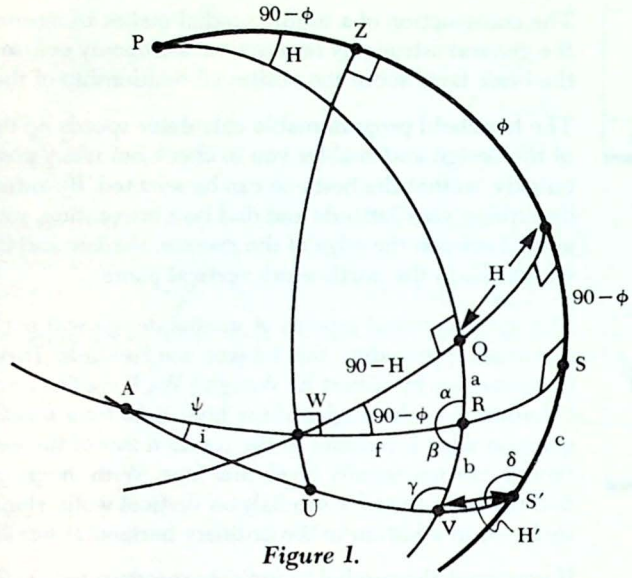


Figure 1.

We want to find H′, the angle between the edge of the gnomon shadow and the line in the dial plane which lies in the north-south vertical plane. The value of H′ will depend on H, ϕ, ψ, and i. The solution is extremely complicated to carry out explicitly, but it can be done on the programmable calculator by just chaining together the numerical solutions of several spherical triangles. In the following outline let CA and CS denote the law of cosines for angles and sides, respectively, and let S denote the law of sines.

In triangle QWR we have, by CA,

$$\cos\alpha = \sin(90°-\phi)\cos(90°-H) \qquad [1]$$

and then, by S,

$$\frac{\sin a = \sin(90°-\phi)\sin(90°-H)}{\sin\alpha} \qquad [2]$$

and, by CS,

$$\cos f = \cos a \cos(90°-H) \qquad [3]$$

Then in triangle ARV, by CA,

$$\cos\gamma = -\cos i \cos\beta + \sin i \sin\beta\cos(\psi+f) \qquad [4]$$

where $\beta = 180°-\alpha$. Also in this triangle, by S,

$$\sin b = \sin i \quad \frac{\sin(\psi+f)}{\sin\gamma} \qquad [5]$$

Now in triangle ASS′, by CA,

$$\cos\delta = \sin i \cos(90°+\psi) \qquad [6]$$

and by S,

$$\sin c = \sin i \left[\frac{\sin(90°+\psi)}{\sin\delta}\right] \qquad [7]$$

Finally, in triangle PVS′, we can find the desired side H′ by CS,

$$\cos H' = \cos(90°+a+b)\cos(180°-\phi+c) + \sin(90°+a+b)\sin(180°+0+c)\cos H \qquad [8]$$

In the program given here, round-off inaccuracies can occasionally cause flashing displays for special values of ψ, i, H, or ϕ (0° or 90°). This problem can be avoided by slightly perturbing the entered values away from these special angles.

THE PROGRAM

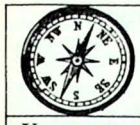
Step #	Program Keystrokes	Purpose/Comments
	OFF/ON LRN	Clears calculator and enters learn mode.
000	2nd Lbl A 2nd Deg STO 0 R/S STO 1 R/S STO 02	Stores ϕ, i, and ψ, in that order.
011	90 − 1 EE 11 +/− = STO 3 × 2 = STO 8 R/S	Stores 90° and 180°. (Note: Subtracting 1×10^{-11} eliminates the round-off inaccuracy mentioned earlier.)
028	2nd Lbl B STO 4 RCL 3 − RCL 0 = 2nd sin STO 13 × (RCL 3 − RCL 4) STO 14 2nd cos = INV 2nd cos STO 5 2nd sin 1/x × RCL 13 × RCL 14 2nd sin = INV 2nd sin STO 6 2nd cos × RCL 14 2nd cos = INV 2nd cos STO 07	Stores H and begins calculation of H′. For triangle QWR, calculates α, a, and f.

Step #	Program Keystrokes	Purpose/Comments
081	RCL 8 − RCL 5 = 2nd cos × RCL 1 2nd cos +/− + RCL 1 2nd sin × (RCL 8 − RCL 5) 2nd sin × (RCL 2 + RCL 7) 2nd cos = INV 2nd cos 2nd sin 1/x × RCL 1 2nd sin × (RCL 2 + RCL 7) 2nd sin = INV 2nd sin SUM 06	For triangle ARV, calculates Y and b, and then sums b to a in register 6.
138	RCL 3 + RCL 2 = 2nd cos × RCL 1 2nd sin = INV 2nd cos 2nd sin 1/x × RCL 1 2nd sin × (RCL 3 + RCL 2) 2nd sin × (1 − 1 EE 11 +/−) = INV 2nd sin STO 09	For triangle AS'S, calculates δ and c.
182	RCL 3 + RCL 6 = STO 10 2nd cos × (RCL 8 − RCL 0 + RCL 9) STO 11 2nd cos + RCL 10 2nd sin × RCL 11 2nd sin × RCL 4 2nd cos = INV 2nd cos INV EE 2nd Prt R/S	Calculates H' (triangle PVS').
	LRN RST	Exits learn mode.

Note: Print command may be omitted if the PC-100A is not used.

REGISTER CONTENTS

R_{00}	ϕ	R_{05}	α	R_{10}	$(90°+a+b)$
R_{01}	i	R_{06}	a, a+b	R_{11}	$(180°-\phi+c)$
R_{02}	ψ	R_{07}	f	R_{12}	
R_{03}	90°	R_{08}	180°	R_{13}	$\sin(90°-\phi)$
R_{04}	H	R_{09}	c	R_{14}	$(90°-H)$



USING THE PROGRAM

Procedure	Press	Display/Comments
Enter data.		
ϕ	A	ϕ
i	R/S	i
ψ	R/S	1.8 02 ($180°-2\times10^{-11}$)
Fix decimal for desired accuracy.	2nd fix n	
Enter H and calculate H′.	B	H′° (Allow several seconds for calculation.)

°This value is printed if the PC-100A is used.

SAMPLE PROBLEM

Given the parameters listed below, calculate H′ for H = 0°, 15°, 30°, 45°, 60°, 75°, and 90°.

Latitude: $\phi = 40°$
Inclination: $i = 90°$ (Vertical wall)
Rotation: $\psi = 20°$ (Facing slightly west of south)

Procedure	Press	Display/Comments
Enter program.		0
Enter data.		
ϕ	40 A	40.
i	90 R/S	90.
ψ	20 R/S	1.8 02
Set decimal for 6 places.	2nd fix 6	1.8000000 02
Enter first value of H and calculate H′.	0 B	0.000115 (degrees)
Repeat for each value of H.	15 B	11.615194
	30 B	22.521310
	45 B	33.450367
	60 B	45.137156
	75 B	58.380070
	90 B	73.987005

11

Sundial Design

Further Explorations

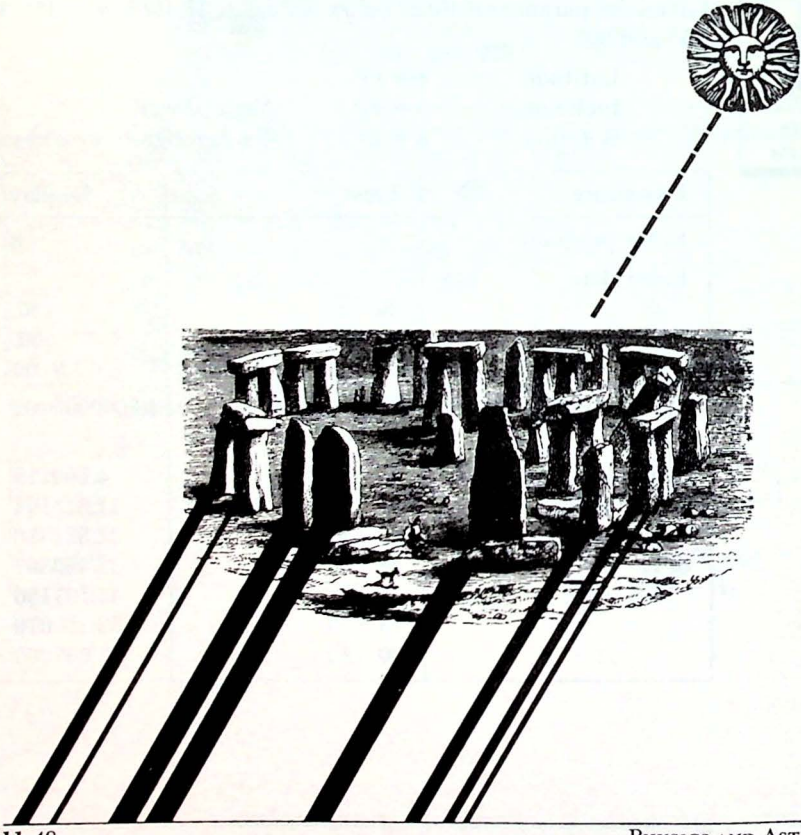
1. Design the hour lines for a sundial at your own latitude and longitude.
2. Design the hour lines for a series of dials mounted on three sides of a level cube aligned to the cardinal directions at your location.
3. Given that the gnomon of a sundial is a flat, wedge-shaped piece attached to the dial face and lying in the North-South vertical plane, what is the angle of the wedge in terms of angles defined in *Figure 1*?

(*Answer:* ϕ - c, where c is given by equations [6] and [7].)

References/ Information

R. N. Mayall and M. L. W. Mayall, *Sundials* (Boston: Hale, Cushman and Flint, 1938).

Albert E. Waugh, *Sundials: Their Theory and Construction* (New York: Dover Publishing Company, 1973).



Index

A
Accelerator, in economy 7-23
Active filter 9-3
Admittance 10-6
Algorithm 1-1, 1-6
Analysis of variance (ANOVA) . 4-31
Arithmetic progression 1-2
Average, mean 4-2

B
Babbage, Charles i-1
Band-pass filter 9-3
Base conversion 1-16
Blood-type
  Distribution in population . 8-14
  Matching 8-8
Bode and Nyquist plots 10-21

C
Capital budgeting 6-12
Cardiac output 9-18
Cash flows 6-5, 6-12
Centered moving average 7-3
Central force motion 11-16
Chords 5-8
Clefs 5-13
Cobb-Douglas production function 7-28
Common base 10-16
Common collector 10-16
Common emitter 10-16
Composition (music) 5-20
  Major and minor scales . . . 5-22
  Pentatonic scales 5-21
  Twelve-tone row 5-23
  Whole-tone scale 5-20
Confidence interval 4-20, 9-7
Convolution 10-39
Correlation coefficient . . . 4-24, 9-30
Cost curves
  Average fixed cost 7-29
  Average total cost 7-29
  Average variable cost 7-29
  Marginal cost 7-29
  Total cost 7-29
Cowell's method 11-16
Curve fitting
  Least-squares approximation 3-26
  Linear regression . . . 4-24, 9-30
Cyclical growth in economy . . . 7-23

D
Degrees of freedom 4-19
Depreciation 6-12
Derivatives, approximation of . . . 3-6
Differential equations 10-44
  Cowell's Method 11-16
  Modified Euler's Method . . 11-8
  Predictor-Corrector Method 8-22
  Runge-Kutta Method 3-14
Diffusion coefficient 8-21
Discrete-time systems response . 10-44
Discriminant 2-15

E
Eccentricity 11-37
Economic Order Quantity (EOQ) . 6-2
ENIAC 8-2
Equation of time 11-37
Error of estimate 9-7
Euclidean algorithm 1-6
Euler's Method, Modified 11-8

F
F-tables 4-38, 4-39
F-test 4-31
Fick equation 9-18
Force 3-23, 11-8, 11-16
Forecasting market shares 6-25
Fraunhofer diffraction 11-2
Functions
  Approximating roots of 3-2
  Graphing ordered pairs . 2-2, 2-22
  Polynomial 2-7
  Speed distribution 11-25
  Transfer 10-21
  Trigonometric 2-22
Fundamental (music) 5-30
Fundamental Theorem of Algebra 1-12
Future value 6-5

G
Gas collisions 11-25
Genetics, Mendelian 8-2
Geometric progression 1-2
Greatest common divisor 1-6
Growth
  Epidemics 8-20
  of economy 7-23
  of population 8-15

H
Harmonics 5-3
Heron's formula 2-19
Hollerith, Herman i-2
Hybrid parameters 10-16
Hypothesis testing 4-20

I
ILLIAC 5-1
Immittance function 10-21
Impedances
  Series-Parallel 10-2
  Smith Chart calculations . . 10-6
Integrated circuit i-2
Integration
  Simpson's rule 4-10, 10-35
  Trapezoidal rule 3-10, 8-22, 10-39
Internal rate of return (IRR) 6-5
Intervals (music) 5-3
Inventory 6-2

J
Jacquard, Joseph Marie i-1

K
Kilby, Jack i-2
Kurtosis 4-5

L
Law of cosines 2-19
Law of sines 2-19
Least common multiple 1-6
Leibniz, Gottfried i-1
Linear equations
  Master Library program . . 2-10
  Row reduction 3-33
Linear regression 4-24, 9-30

M
MARK I i-2
Markov analysis 6-25
Mean, arithmetic 4-1
Melodic contours 5-13
Motion
  Equation of 8-21
  Studies of 3-23, 11-8
Multiplier, Keynesian 7-23
MUSIC IVB 5-1

INDEX

N

Newton-Cotes formula 3-15
Newton-Raphson method . . .3-1, 6-5
Newton's law of cooling 3-21
Normal curve 4-10
Null hypothesis 4-20, 4-25
Number of samples, estimating . . 9-7
Nyquist - plot data 10-21

O

Obliquity 11-37
Orbits
  About the earth 11-37
  Inverse square law 11-21

P

Partials (music) 5-30
Pascal, Blaise i-1
Passband filter 9-3
Pendulum 11-15
Perfect number 1-15
Pitch frequency 5-26
Polynomial
  Curve fitting 3-26
  Evaluation 2-7
Present value 6-5
Prime factors 1-12
Prime number testing 1-9
Programming techniques i-4
Punnett square 8-2

Q

Quadratic equations 2-15
Queuing 6-20

R

Ratio to centered average 7-3
Rectangular pulse response . . .10-35
Reference cycle 7-9
Reflection factor 10-6
Root-Locus 10-28
Runge-Kutta method 3-14

S

Safety stock 6-2
Sample size determination 9-7
Seasonal index 7-3
Sedimentation velocity 8-21
Sequences 1-2
Series 1-2
Service facilities, least cost 6-20
Short-circuited line 10-7
Sieve of Eratosthenes 1-9
Single-stub impedance match . . 10-8
Simpson's approximation 10-35
Skewness 4-6
Slotted-line measurement 10-6
Smith Chart 10-6
Speed distribution,
  two-dimensional 11-25
Spherical triangles 11-38, 11-43
Standard deviation 4-2
Standard error 9-7
Standard normal curve 4-15
Stock-out 6-2
Stub length 10-8
Sundials 11-43

T

t-distribution 4-19
t-tables 4-36, 4-37
t-test 4-25, 9-24
Traits, dominant and recessive . . . 8-2
Transducer scaling 9-10
Transfer function 10-21
Transistor parameter conversion 10-16
Transistors i-2
Transition probability 6-25
Transmission line 10-6
Trapezoidal rule 3-10, 8-22
Trend line 7-18

U

Unit step response 10-35

V

Variable periodic cash flows 6-5
Variance 4-4
Voltage minimum 10-6
Voltage standing wave
  ratio (VSWR) 10-6

X, Y, Z

z-scores 4-10, 4-15

Programmable Calculators...

Place the power that used to be exclusively the domain of large computers right into the palm of your hand. At your desk, in class, in the lab, in the field – you now have instant access to an accurate, easy-to-use computational system. New dimensions in analysis, decision-making, and learning are opened up for you, with "on-line" power and in-hand convenience.

Sourcebook for Programmable Calculators...

Goes far beyond any owner's manual in exploring the applications of the programmable calculator in your field of study or profession. Step-by-step examples and programs give you an in-depth look at the way machines like the *TI Programmable 58*™ and the *TI Programmable 59*™ calculators can be applied as an integral part of classroom study or professional work. Fast, accurate calculation, rapid data tabulation and graphing, exploration of mathematical formulas and models, quick statistical analysis and projection – these are just part of what's possible with the programmable calculator. It can give you the edge you just can't afford to be without in today's fast-paced world.

Check These Topics:

- Basic Number Theory
- College Algebra/Trigonometry
- Calculus/Linear Algebra
- Statistics
- Music Theory
- Business/Operations Research
- Economics
- Biology
- Biomedical Engineering
- Electrical/Systems Engineering
- Physics/Astronomy

If you already own a programmable calculator...

This book can open up far more of its power for you with applications in areas you may never have even considered.

If you don't own one...

This book shows you just how easy-to-use, powerful – and necessary – a programmable calculator can be as part of your studies or career.

TEXAS INSTRUMENTS
INCORPORATED

LEARNING
CENTER

PRINTED IN USA

1015029-1